

## THESIS / THÈSE

### MASTER EN SCIENCES INFORMATIQUES

#### Boîte à outils pour jeux de stratégie

Dubois, Yves

*Award date:*  
2002

[Link to publication](#)

#### General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal ?

#### Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

**Facultés Universitaires Notre-Dame de la Paix, Namur**

**Institut d'informatique**

**Année académique 2001-2002**

**Boîte à outils  
pour jeux de stratégie**

**Yves Dubois**

**Mémoire présenté en vue de l'obtention du grade de  
Licencié en Informatique**

## **Résumé**

Nous avons voulu développer dans le cadre de ce mémoire un jeu de stratégie générique, aisément adaptable à des contextes de jeu variés et extrêmement modulable. Outre la discussion des aspects purement ludiques et des concepts propres à ce type de jeu, nous avons tenté de mettre l'accent sur la polyvalence et la portabilité de notre application java, notamment grâce à une architecture logique, basée sur la complémentarité interfaces et classes abstraites/classes concrètes, et au recours systématique aux classes archétypales de Sun (particulièrement en matière d'interface graphique). Afin de démontrer non seulement la généricité du programme mais aussi son caractère réellement ludique, nous avons implémenté un jeu exemplatif –minimaliste- ayant pour thème la guerre du Pacifique. Même si notre approche n'a pas pu être aussi systématique ni aussi complète que nous l'aurions voulu –certains de nos a priori s'étant révélés peu fondés-, nous espérons avoir mis en évidence plusieurs concepts propres à ce type de programmation et fourni une base de travail exploitable pour d'éventuels mémoires désireux d'approfondir tel ou tel aspect en rapport avec ce sujet.

Mots-clés : jeu, stratégie, réseau, pion, java.

## **Abstract**

With this thesis, we wanted to develop a generic war game, easily convertible to various game universes and highly modifiable. Next to a study of some specific gaming concepts, we tried to emphasize both generality and portability of our java application, mainly thanks to a logical architecture based on the complementing interfaces, abstract and real classes, and by systematically using the archetypal Sun classes (mainly for the graphical interface). We decided to implement a minimal example game centered on the Pacific War to show not only the program's generality but also its real gaming capabilities. Even if our development wasn't as systematic and thorough full as was originally planned –some of our preconceptions being found misguided-, we hope we hereby managed to pinpoint several peculiar concepts often found in such kind of programs and to ready a working basis for potential future works furthering some of the aspects we so briefly described in our work.

Keywords : game, strategy, network, pion, java.

## **Avant-propos**

Nous tenons particulièrement à remercier ici notre promoteur, Monsieur Leclercq, pour sa disponibilité et son aide tant dans la définition de l'orientation de notre mémoire que dans la structuration et le développement de celui-ci. Nous adressons aussi nos remerciements les plus sincères à nos amis et parents pour leurs avis techniques fort appréciés et pour avoir mené à bien la longue et pénible tâche de relecture de notre texte. Enfin, nous remercions vivement les membres du kot-à-projet « Le kot de mailles » de Louvain-la-Neuve pour leur intérêt soutenu et la confiance qu'ils nous ont témoignée en utilisant effectivement notre logiciel dans le cadre de leur jeu de stratégie géant « Hégémonie ».

# Table des Matières

<b><u>RÉSUMÉ</u></b>	<b>1</b>
<b><u>ABSTRACT</u></b>	<b>1</b>
<b><u>AVANT-PROPOS</u></b>	<b>2</b>
<b><u>TABLE DES ILLUSTRATIONS</u></b>	<b>6</b>
<b><u>TABLEAUX</u></b>	<b>7</b>
<b><u>INTRODUCTION GÉNÉRALE</u></b>	<b>8</b>
<b><u>I. 1<sup>ÈRE</sup> PARTIE : CADRE ET CONCEPTS</u></b>	<b>10</b>
<b><u>I - 1. Introduction</u></b>	<b>10</b>
<b><u>I - 2. Discussion générale</u></b>	<b>10</b>
<b><u>I - 3. Concepts et acceptions</u></b>	<b>11</b>
<b><u>I - 4. Buts et définition</u></b>	<b>14</b>
I - 4.1. Perception de l'utilisateur	15
I - 4.2. Mécanismes de jeu	16
I - 4.3. Transmission de l'information	17
<b><u>I - 5. Conclusion</u></b>	<b>17</b>
<b><u>II. 2<sup>ÈME</sup> PARTIE : CONCEPTUALISATION</u></b>	<b>18</b>
<b><u>II - 1. Introduction</u></b>	<b>18</b>
<b><u>II - 2. Objectifs et particularisations</u></b>	<b>18</b>
II - 2.1. Objectifs particuliers	18
II - 2.2. Perception de l'utilisateur	18
II - 2.3. Mécanismes de jeu	19
II - 2.4. Transmission de l'information	22
<b><u>II - 3. Concepts</u></b>	<b>22</b>
II - 3.1. Perception de l'utilisateur	22
II - 3.2. Mécanismes de jeu	31
<b><u>II - 4. Conclusion</u></b>	<b>35</b>
<b><u>III. 3<sup>ÈME</sup> PARTIE : INFORMATISATION</u></b>	<b>36</b>

<b><u>III - 1.</u></b>	<b><u>Introduction</u></b>	<b>36</b>
<b><u>III - 2.</u></b>	<b><u>Concepts informatiques</u></b>	<b>36</b>
III - 2.1.	Objets principaux	36
III - 2.2.	Objets secondaires (regroupement)	38
III - 2.3.	Résumé architectural	38
<b><u>III - 3.</u></b>	<b><u>Questions d'informatisation</u></b>	<b>39</b>
III - 3.1.	Structuration générale	39
III - 3.2.	Choix d'implémentation	40
III - 3.3.	Spécifications réduites :	46
<b><u>III - 4.</u></b>	<b><u>Concepts</u></b>	<b>50</b>
III - 4.1.	Perception de l'utilisateur	50
III - 4.2.	Mécanismes de jeu	54
III - 4.3.	Transmission de l'information	56
<b><u>III - 5.</u></b>	<b><u>Conclusion</u></b>	<b>57</b>
<b><u>IV.</u></b>	<b><u>4<sup>ÈME</sup> PARTIE : IMPLÉMENTATION</u></b>	<b>58</b>
<b><u>IV - 1.</u></b>	<b><u>Introduction</u></b>	<b>58</b>
<b><u>IV - 2.</u></b>	<b><u>Concepts</u></b>	<b>58</b>
IV - 2.1.	Détermination du chemin	58
IV - 2.2.	Remote Control	64
IV - 2.3.	Editeur de zones/terrains	64
<b><u>IV - 3.</u></b>	<b><u>Structure et découpage en packages</u></b>	<b>65</b>
IV - 3.1.	Package ygwan	65
IV - 3.2.	Package ygwan.grid	67
IV - 3.3.	Package ygwan.network	68
IV - 3.4.	Package ygwan.pions	70
IV - 3.5.	Package ygwan.swing	71
IV - 3.6.	Package ygwan.swing.dialogs	74
IV - 3.7.	Package ygwan.swing.listeners	76
IV - 3.8.	Package ygwan.test	76
IV - 3.9.	Package ygwan.util	78
IV - 3.10.	Package ygwan.world	79
IV - 3.11.	Exemple de code	79
<b><u>IV - 4.</u></b>	<b><u>Conclusion</u></b>	<b>82</b>
	<b><u>CONCLUSION GÉNÉRALE</u></b>	<b>83</b>
	<b><u>Objectifs atteints</u></b>	<b>83</b>
	<b><u>Difficultés et apprentissage</u></b>	<b>83</b>
	<b><u>Améliorations futures</u></b>	<b>83</b>

<b><u>ANNEXES</u></b>	<b>85</b>
<b><u>Annexe 1 :</u></b>	<b>85</b>
<b><u>La stratégie dans l'histoire</u></b>	<b>85</b>
<u>Classification des niveaux d'abstraction</u>	85
<u>Classification des stratégies militaires</u>	87
<u>Les sept manœuvres classiques de la guerre :</u>	90
<b><u>Annexe 2 :</u></b>	<b>96</b>
<b><u>La stratégie japonaise lors de la guerre du Pacifique</u></b>	<b>96</b>
<b><u>BIBLIOGRAPHIE</u></b>	<b>109</b>
<b><u>Informatique</u></b>	<b>109</b>
<b><u>Deuxième guerre mondiale</u></b>	<b>109</b>
<b><u>Stratégie</u></b>	<b>110</b>

## Table des illustrations

Figure 1 : exemple classique du jeu de dames.....	13
Figure 2 : propriétés de terrain pixel par pixel.....	23
Figures 3 : propriétés de terrain par zones homogènes .....	23
Figure 4 : système par case .....	24
Figure 5 : numérotation des sommets d'un hexagone .....	25
Figure 6 : découpe T0-C0.....	26
Figure 7 : découpe H0-H3 .....	26
Figure 8 : découpe R0-L0 .....	27
Figure 9 : découpe Q0 .....	27
Figure 10 : exemple de rotation .....	28
Figure 11 : état de jeu de l'exemple.....	30
Figure 12 : mécanismes de filtre et vision résultante .....	31
Figure 13 : gestion d'une pile de pions .....	32
Figure 14 : architecture générale .....	39
Figure 15 : architecture logique .....	44
Figure 16 : mécanisme de transfert asynchrone.....	48
Figure 17 : mécanismes de l'interface .....	49
Figure 18 : interface avec traitement « local » .....	49
Figure 19 : Vue globale de l'interface initiale .....	50
Figure 20 : découpe de l'interface .....	51
Figure 21 : les différentes couches de l'interface.....	52
Figure 22 : mécanismes de filtre.....	53
Figure 23 : système de génération des données.....	55
Figure 24 : schéma de fonctionnement de ClientStub .....	57
Figure 25 : Coordonnées relatives des hexagones .....	59



Figure 26 : établissement d'un chemin avec génération du parcours .....	61
Figure 27 : établissement d'un chemin case à case .....	62
Figure 28 : schéma logique du remote control.....	64
Figure 29 : fonctionnement de l'observer .....	69
Figure 30 : découpage en classes de l'interface.....	71
Figure 31 : Vue de l'interface générale .....	71
Figure 32 : structure des classes des boutons .....	72
Figure 33 : Construction du JHQPannel .....	73
Figure 34 : JServerFrame .....	74
Figure 35 : JServerDialog .....	75
Figure 36 : Menu de sélection de fichiers .....	75

## **Tableaux**

Tableau 1 : Découpes des hexagones .....	28
Tableau 2 : Propriétés des pions .....	33
Tableau 3 : Propriétés des hexagones .....	34
Tableau 4 : Coordonnées relatives des hexagones.....	59

## Introduction générale

De tous temps, les jeux ont représenté une part importante de l'activité humaine ; aujourd'hui plus encore que par le passé ils font partie intégrante de notre mode de vie au point de justifier le développement d'une véritable industrie. Les jeux informatiques constituent un des domaines dont l'évolution est la plus dépendante des développements techniques, et ce parfois -nous semble-t-il- au détriment de la profondeur et de l'intérêt ludique. Sans prétendre en aucune manière apporter une approche novatrice à ce domaine fort professionnalisé, nous espérons néanmoins réaliser dans ce travail l'ébauche d'un vrai jeu de stratégie et ce d'une manière efficace et élégante. A ce titre, notre attention se portera surtout sur la généricité du programme développé et sa portabilité au détriment parfois d'une plus grande complexité de scénario.

L'étendue et la complexité d'un sujet touchant à des domaines informatiques variés nous ont obligés à opérer un certain nombre de choix, parfois difficiles, afin de concilier des exigences contradictoires de complétude et de complexité ; nous avons ainsi décidé de privilégier le développement de l'ensemble des aspects du jeu au détriment du traitement plus approfondi de certains domaines, tout en laissant malgré tout de côté des problèmes intéressants mais qui nécessiteraient presque un mémoire à eux seuls : ainsi, nous n'étudierons pas la définition et l'implantation d'une stratégie par un joueur artificiel.

Pour réaliser ce travail, nous avons eu recours à des sources diversifiées qui se sont révélées plus ou moins utiles selon les parties abordées. Ainsi, pour la partie plus conceptuelle de ce mémoire, nous avons pu consulter une série d'ouvrages ou de sites web assez complets sur les choix d'informatisation. De même, nous avons consulté plusieurs textes de référence pour la réalisation de certaines de nos annexes. Cependant, d'autres sections, comme le développement de concepts propres à notre objet par exemple, sont essentiellement nées d'un processus de réflexion propre auquel nous avons été encouragés par la rareté ou le peu de pertinence de l'information disponible.

Pour la partie plus concrète et informatique, outre la documentation disponible sur le site de Sun<sup>1</sup>, des exemples concrets ont parfois contribué à nous inspirer certains choix de réalisation ou de structure tandis que d'autres aspects de l'implémentation ont dû être réalisés indépendamment de tout modèle. Enfin, les données servant à notre exemple concret ont dû être créées spécifiquement à cette fin (carte du Pacifique, propriétés des cases, propriétés des pions, documents d'aide et de contextualisation, etc).

Pour la présentation de nos résultats, nous avons fait le choix d'une découpe par niveaux d'informatisation, du plus général au plus technique ; nous aborderons ainsi successivement quatre « niveaux » de concrétisation de notre programme.

---

<sup>1</sup> [http://java.sun.com/j2ee/sdk\\_1.3/techdocs/api/index.html](http://java.sun.com/j2ee/sdk_1.3/techdocs/api/index.html).

D'abord, dans une première partie, nous tenterons de comprendre ce qu'est un jeu de stratégie, ses caractéristiques et les objectifs de ce genre de jeu ; nous définirons également un certain nombre de concepts généraux utilisés dans l'ensemble de ce travail.

Ensuite, dans une seconde partie, nous dégagerons les particularités non informatiques du jeu de stratégie que nous voudrions réaliser et les objectifs généraux que nous poursuivons ce faisant. Après avoir précisé le modèle de jeu que nous désirons développer et les actions réalisables par les joueurs, nous évoquerons un certain nombre de thèmes de réflexion utiles à la concrétisation des concepts définis dans la première partie.

Par après, dans une troisième partie, nous aborderons les questions d'informatisation, les contraintes qu'elles imposent ou les possibilités qu'elles nous offrent. Nous définirons alors une série de concepts, version informatique de certaines idées de la première partie ou nouveaux objets. Nous poserons enfin une série de choix informatiques, justifiés ou arbitraires.

Enfin, dans une quatrième partie, nous aborderons l'implémentation à proprement parler, la structure de notre programme, la hiérarchisation des objets de jeu et étudierons particulièrement une classe exemplative.

Une conclusion générale en trois axes nous permettra successivement d'évaluer les réalisations accomplies au terme de ce travail, de préciser les limitations choisies ou involontaires et les connaissances que nous en avons retirées et finalement les perspectives de développement et d'amélioration.

Une section d'annexes précédera une bibliographie regroupant les sources utilisées lors de la rédaction de ce travail ; annexe 1, la stratégie, mot et réalité ; annexe 2, la guerre du Pacifique du côté japonais, stratégie ou folie ? Enfin, vous pourrez prendre connaissance du code de notre programme dans un document séparé.

# **I. 1<sup>ère</sup> partie : Cadre et concepts**

## ***I - 1. Introduction***

Dans cette première partie, nous tenterons de rendre le cadre général de notre mémoire perceptible au lecteur en clarifiant ce que nous entendons par un jeu de stratégie ; pour ce faire, nous subdiviserons notre développement en trois sections. Dans un premier temps, nous entreprendrons de présenter l'objet de notre mémoire de la façon la plus générale, discutant notamment des concepts de jeu, de stratégie et de la catégorie très particulière des jeux de stratégie. Dans un second temps, nous tenterons de clarifier certains termes et notions dont la compréhension nous apparaît essentielle et de préciser le sens que nous voulons leur donner dans le cadre de ce travail. Dans un troisième temps enfin, nous présenterons notre vision personnelle des éléments devant être au départ de toute conception d'un jeu de stratégie général.

## ***I - 2. Discussion générale***

De tous temps, le jeu a occupé une place essentielle dans les activités humaines : dès l'origine, il a présenté les caractères d'ambivalence qui le caractérisent aujourd'hui : à la fois délassément mais aussi lieu d'entraînement de comportements par ailleurs essentiels -ou simplement utiles- à la survie ou au développement de l'individu au sein de la société. Sans même prendre en compte les aspects sociaux et culturels, souvent historiquement liés à ces pratiques que nous qualifierions de ludiques, l'importance de ce pan des activités humaines ne s'est jamais démentie, quelles que soient les formes qu'il ait pu adopter selon les lieux et les époques.

Dans notre société actuelle, souvent qualifiée de société des loisirs, le développement d'une réelle industrie se consacrant intégralement à la création ou à l'organisation de grands événements sportifs ou ludiques n'a pas de quoi surprendre. Il est cependant intéressant de constater la diversité des genres de jeux et leurs importances relatives ; ainsi, alors que certaines catégories de jeux sont très communément attestées, d'autres sont bien plus discrètes : parmi ces dernières les jeux de stratégie.

Parallèlement, le terme de stratégie<sup>2</sup> semble aujourd'hui être présent dans de multiples domaines : guerre bien sûr, mais aussi économie, politique ou activités sociales sont dorénavant des champs courants d'application de « stratégies ». Mais si ce terme semble être dorénavant tombé dans le domaine public, il connaît une efflorescence encore plus marquée dans le domaine ludique. Il existe bien sûr de nombreuses définitions de ce qui constitue une stratégie à part entière dans ces

---

<sup>2</sup> Voir annexe 1 pour une discussion de ce terme.

différents domaines mais il n'en existe guère qui définisse les caractéristiques attendues d'un jeu pour qu'il soit qualifié de « jeu de stratégie ».

Le but de cette première partie étant de nous permettre de préciser cette notion, nous nous contenterons pour l'instant d'une définition généraliste et des plus minimalistes : un jeu de stratégie est un jeu qui fait intervenir une stratégie.

Ainsi définis, l'histoire des jeux de stratégie (ex : échecs, go, etc) remonte elle aussi à la plus lointaine antiquité et n'a cessé d'occuper une place importante dans les loisirs et l'éducation, même si l'émergence d'autres types de jeux « de société » ou encore « de table » a eu tendance à diminuer son importance relative au cours de l'histoire récente.

Plus récemment encore (histoire moderne), les progrès de la stratégie militaire - principalement dans ses aspects théoriques et « simulationnistes » - a mené à l'émergence de nouveaux jeux de stratégie - au sens restreint qui nous intéressera ici - dont les *kriegsspielen*<sup>3</sup> de l'école allemande constituèrent en leur temps l'exemple le plus abouti.

Mais c'est au cours de ces trente dernières années que le genre s'est diversifié pour sortir des traditionnels affrontements de soldats de plomb pour conquérir des zones de l'économie autrement plus réelles que toute table de jeu. On peut aujourd'hui trouver des jeux de stratégie - ou encore wargames selon l'appellation anglophone – concernant une grande diversité de conflits historiques ou imaginaires, offrant aux amateurs des défis sans cesse renouvelés par les variantes de règles et nouveaux systèmes de jeu. Si ce genre de jeu connaît toujours un moins grand succès populaire que d'autres catégories, il n'en est pas moins devenu un secteur rentable<sup>4</sup>.

### **I - 3. Concepts et acceptions**

Afin de nous aider à définir les caractéristiques attendues d'un jeu de stratégie, il nous semble important de clarifier certaines des notions que nous emploierons tout au long de ce travail et d'éviter toute mauvaise compréhension de celles-ci :

---

<sup>3</sup> Ou comme le nom l'indique « jeux de guerre » : il s'agissait d'exercices stratégiques d'un grand réalisme (se déroulant sur de vastes tables aménagées) incluant la gestion d'un grand nombre de facteurs que l'Etat major estimait incontournables dans la réalité et visant à entraîner et à mettre en situation les futurs officiers de l'armée prussienne (puis allemande).

<sup>4</sup> Ainsi, en matière de jeu de stratégie ayant pour cadre un univers fantastique au sens littéraire du terme, le succès de Games Workshop et son orientation essentiellement commerciale ne sont plus à démontrer.

#### *a) Mécanismes de jeu (tour par tour, ordres simultanés, temps réel)*

Les mécanismes peuvent différer fortement d'un jeu de stratégie à l'autre. Le modèle le plus ancien et le plus « classique » est le tour par tour où les joueurs passent leurs ordres l'un après l'autre en fonction d'une convention de passage de tour (temps imparti, quantité d'ordres, échec d'une action, etc). Un second mécanisme a fait son apparition par la suite afin d'ajouter plus de réalisme au jeu en simulant la méconnaissance des décisions de l'adversaire sans recourir pour autant à un jeu en aveugle : le mécanisme par ordres simultanés où deux phases de jeu se succèdent : celle du passage d'ordres par les joueurs et celle de la résolution simultanée de ceux-ci. Enfin, un troisième et dernier mécanisme -dit en temps réel- autorise les joueurs à passer leurs ordres sans aucune découpe en phase des actions.

Les exemples d'utilisation de chacun de ces mécanismes abondent dans l'univers des jeux de stratégie ; s'ils demandent une rapidité de réaction et présentent une simultanéité d'action croissante du premier au dernier, la complexité inhérente au jeu a souvent tendance à se réduire en proportion. Dans le cas d'un jeu de stratégie transposé sur ordinateur, la question s'accroît d'un volet informatique puisque ce choix de mécanisme peut conditionner une partie des structures du programme mais aussi, de façon plus évidente, le séquençage des actions de jeu.

#### *b) Tour*

Concept de structuration du temps, le tour permet un séquençage des actions des joueurs dans la plupart des jeux de stratégie sur table. Le but principal est de simplifier l'aspect temporel du passage d'ordres et d'éviter les incertitudes liées à celui-ci par un cloisonnement strict et l'utilisation de mécanismes de sélection du joueur actif courant. La fin du tour de ce joueur en faveur du début de tour d'un autre joueur fait appel à des mécanismes divers, depuis la fin volontaire du tour du joueur actif à l'instauration d'une durée de tour maximum en passant par les conséquences d'actions entreprises.

#### *c) Evaluation et gain*

Dans tout jeu de stratégie, on désire pouvoir évaluer la pertinence des stratégies ou tactiques adoptées afin d'acquérir une plus grande expérience des choses ; la théorie des jeux est un domaine assez complexe que nous n'aborderons pas ici ; nous n'en retiendrons que les concepts de base et particulièrement celui de jeu à somme nulle où l'addition des points de victoire d'un des adversaire compense la perte de points de son/ses adversaire(s). D'autres critères plus fins pourraient être introduits avec profit mais au détriment de la simplicité de l'ensemble. Dès lors, nous considérerons que si la destruction d'unité ennemie rapporte 100 points de victoire, par exemple, la perte de cette même unité coûtera 100 points au joueur concerné. Ainsi, la somme totale des points de victoire de l'ensemble des joueurs sera toujours égale à 0. L'utilisation de ce simple mécanisme entraîne que pour gagner (sur base du seul critère des unités), un joueur doit détruire plus d'unités qu'il n'en perd, ce qui ne semble pas absurde dans un conflit équilibré.

#### *d) Pion*

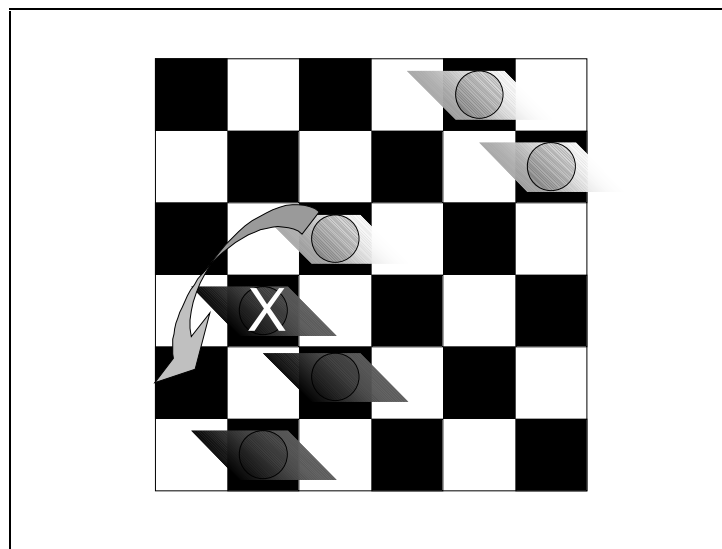
Un pion est un objet servant de représentation consensuelle d'éléments - généralement mobiles- de la réalité particulière modélisée dans le jeu. Dans le cas des jeux de stratégie, il s'agira le plus souvent d'unités militaires de taille variable, souvent déterminée par l'échelle de la représentation du terrain d'opération, et différenciées par des caractéristiques propres.

#### *e) Case*

Une case est l'élément atomique du plateau de jeu ; sa taille comme son nombre d'occurrence peuvent grandement varier selon le jeu. De la même manière que pour les pions, les dimensions de la réalité représentée peuvent être extrêmement variables.

#### *f) Bataille*

La disposition spatiale des pions sur les cases est à la base de tout jeu de stratégie ; dès lors, le jeu progresse vers sa fin grâce à l'établissement de conditions de victoire. Un événement notable dans cette progression survient fréquemment lorsque deux pièces appartenant à des joueurs ennemis se trouvent dans une disposition spatiale donnée. Selon le mécanisme le plus fréquemment rencontré, lorsque deux pièces ennemies occupent la même case, une bataille a lieu, entraînant des conséquences pouvant aller de la suppression d'un des pions du terrain de jeu à un repositionnement de certains pions ou à une simple modification des éléments de comptabilisation des conditions de victoire.



**Figure 1 : exemple classique du jeu de dames**

#### *g) Règles*

Les règles sont des limitations des possibilités d'action des joueurs au cours d'une partie sur lesquels tous se mettent généralement d'accord avant le début de

celle-ci et supposées aider à modéliser des caractéristiques du réel représenté ou d'assurer un certain équilibre entre les participants.

#### *h) Partie*

Une partie est l'ensemble des actions effectuées par les joueurs entre le début de leur interaction et la fin de celle-ci. Ce concept est particulièrement flou dans la mesure où il relève principalement de l'acception des parties en présence, mais en règle générale, on peut plus précisément le définir comme la succession des mouvements élémentaires séparant une disposition initiale d'une disposition finale (i.e. répondant à un ou plusieurs critères de terminaison).

#### *i) Joueur*

Un joueur est l'un des intervenants du jeu, souvent caractérisé par la distinction fonctionnelle établie entre les groupes de pions. A priori, il constitue l'unité atomique des camps en présence, même si un joueur peut être constitué de plusieurs individus ou au contraire deux joueurs n'être qu'un seul individu.

#### *j) Camp*

Un camp est la réunion des pions des joueurs partageant des objectifs communs, soit réels, soit supposés. Généralement, au moins deux camps s'affrontent. Même si des jeux à un seul camp existent, ils présupposent souvent l'existence d'un camp non explicité regroupant les facteurs et règles de jeu.

#### *k) Unité*

Une unité désigne généralement tout ensemble organisé au sein d'une hiérarchie ; par restriction, nous définirons une unité dans le cadre de ce mémoire comme l'entité élémentaire de chaque type de troupe : par exemple, un fantassin, un cuirassé, etc.

### **I - 4. Buts et définition**

Pour tenter de préciser les buts de tout jeu de stratégie général, il est primordial de définir ce qu'on entend par ces derniers termes ; en effet, nous avons proposé dans notre présentation générale une définition peu aboutie ne pouvant guère nous aider en la matière.

Il nous apparaît en réalité difficile de proposer une définition précise du concept de jeu de stratégie, tant les principes mêmes diffèrent d'un jeu à l'autre ; cependant, sur base des caractéristiques relevées dans les jeux de ce genre, nous pouvons avancer la définition fonctionnelle suivante : un jeu de stratégie est un jeu de réflexion portant sur le déplacement ou l'action de pions, dans un environnement aux règles établies, mimant avec plus ou moins de réalisme une situation réelle et présentant des conséquences finales imprévisibles ( i.e. non déterministe pour



l'homme, soit en raison de mécanismes de hasard, soit en raison du très vaste domaine des solutions possibles).

Mais cette définition, si elle présente un net progrès par rapport à notre compréhension antérieure de ces termes, concerne toujours une classe de jeux très générale nous empêchant de préciser plus avant notre propos. Une nouvelle spécialisation de notre définition apparaît donc nécessaire, pour ne plus considérer que le sous-groupe des « vrais » jeux de stratégie tel que nous l'avons évoqué plus haut. Ces jeux se particularisent essentiellement par un souci de réalisme plus poussé, un côté guerrier plus affirmé et le désir de proposer un vaste choix d'options aux joueurs.

Les caractéristiques de notre jeu idéal semblent donc consister principalement en critères de réalisme, obtenus par la gestion de paramètres multiples, et permettant la mise en oeuvre de stratégies variées, limitant le moins possible l'imagination et obligeant à la réflexion dans la création, puis la mise en oeuvre d'une stratégie, susceptible d'être remise en question à tout moment par les actions du ou des autres joueurs.

De cette définition découlent de façon naturelle les objectifs principaux que nous devons tenter d'atteindre durant le développement de notre jeu de stratégie. Nous pouvons catégoriser ces questions en trois grands ensembles :

- la perception de l'utilisateur ;
- les mécanismes de jeu ;
- la transmission de l'information.

#### **I - 4.1. Perception de l'utilisateur**

##### *a) Structuration de l'information*

L'espace de jeu doit être conçu de manière à être aisément appréhensible dans sa totalité, les camps comme les types d'unités nettement différenciés, les caractéristiques particulières le plus aisément détectables, etc

##### *b) Aides de jeu*

Des aides de jeu de qualité, les plus complètes possible, doivent permettre au joueur de compléter sa connaissance du jeu à tout moment, tant sur l'état de la partie en elle-même, le statut d'un pion ou ses caractéristiques que sur les mécanismes de jeu les plus divers.

##### *c) Facteurs cognitifs*

Il faut prêter attention au facteur cognitif afin de ne pas surcharger le joueur d'informations non exploitables ou compliquées à l'extrême. Pour ce faire, on doit

faire particulièrement attention au nombre de pions en action, à l'utilisation de mécanismes de jeu intuitifs, à l'emploi de représentations classiques, au choix des différences entre pions, camps, propriétés, etc

## **I - 4.2. Mécanismes de jeu**

### *a) Réalisme des règles*

Les règles doivent être les plus complètes possible, non pas pour rechercher la complexité elle-même mais pour pouvoir simuler avec le plus de finesse possible les événements du réel que l'on veut mettre en scène.

### *b) Corps de règles cohérent et bien pensé*

Ces règles se doivent d'être développées en tant que système, sans contradictions internes ou cas prêtant au doute. Elles doivent être d'application aisée, et ne pas demander un apprentissage trop long aux joueurs (la durée exacte de l'apprentissage nécessaire peut cependant varier grandement à l'intérieur de ces limites conceptuelles).

### *c) Options étendues*

Les règles doivent pouvoir être ajustées ou modifiées par les joueurs avec une grande facilité, afin de coller à leur style de jeu propre ou au genre de conflit qu'ils désirent modéliser. Un des aspects majeurs du jeu doit donc être sa flexibilité et l'abstraction de quelques éléments clés du système de jeu afin de permettre leur adaptation à des contextes très divers.

### *d) Multiplicité des stratégies*

Il doit exister un ensemble le plus vaste possible de stratégies menant à la victoire ; le facteur discriminant ne doit pas être un mécanisme de règle mais les stratégies employées par le ou les adversaires pour contrer telle ou telle stratégie.

### *e) Diversité des facteurs de jeu*

Le jeu doit présenter un nombre de facteurs de jeu suffisamment élevé pour éviter l'existence de stratégies « parfaites » ne pouvant être contrées quels que soient les moyens mis en œuvre par l'adversaire. Ce point est grandement fonction des conditions de victoire choisies et doit donc être pensé conjointement à celles-ci.

### *f) Particularisation des objectifs de victoire*

Les objectifs de victoire, tout comme les facteurs de jeu, doivent être les plus diversifiés ou ouverts possible afin de permettre des styles de jeu très différents selon le modèle de jeu choisi. Ces objectifs sont une part essentielle du jeu dans la

mesure où ils conditionneront pour une part l'efficacité des stratégies employées en déterminant l'importance relative de plusieurs des facteurs de jeu.

### **I - 4.3. Transmission de l'information**

#### *a) Richesse des mécanismes*

Les mécanismes de transmission de l'information doivent être les plus riches et les plus étendus possible afin de permettre divers degrés de réalisme et de style de jeu. Evidemment, plus ces mécanismes seront complexes, plus la vision du jeu qu'en aura le joueur sera éloignée de la réalité. L'imagination et l'expérience du joueur prendront alors le pas sur la simple réflexion en fonction de facteurs connus et maîtrisés.

#### *b) Particularisation des transmissions d'information*

Ces mécanismes doivent pouvoir être modifiés de multiples manières, encore une fois afin de permettre des styles de jeu aussi variés que possible. L'abstraction des concepts essentiels de transmission sera ici aussi un puissant moyen de faciliter cette diversité.

### **I - 5. Conclusion**

Les jeux de stratégie, malgré leurs différences, présentent de nombreuses similitudes, qui nous permettent d'espérer réaliser un jeu générique, pouvant être facilement modifié pour couvrir des styles de jeu divers. Nous avons dans cette première partie défini avec plus de précision les concepts fondateurs de ce genre ludique et explicité l'acception à donner aux termes spécialisés que nous emploierons. Cette formalisation des aspects génériques constitutifs de tout jeu de stratégie nous amène à présent à nous interroger sur les spécificités du moteur de jeu que nous souhaitons concevoir.

## **II. 2<sup>ème</sup> partie : Conceptualisation**

### ***II - 1. Introduction***

Dans un premier temps, nous essayerons de définir de façon claire et détaillée les objectifs que nous poserons et les particularisations que nous choisirons lors de la concrétisation de notre jeu au départ des objectifs d'un jeu de stratégie général. Dans un deuxième temps, nous tenterons de présenter nos réflexions sur un certain nombre de concepts afin d'explicitier notre réflexion.

### ***II - 2. Objectifs et particularisations***

#### **II - 2.1. Objectifs particuliers**

Dans le processus de conception d'un mécanisme de jeu, de nombreux choix restrictifs s'imposent ; comme nous désirons fermer le moins de portes possible, nous allons tenter de conserver le plus de généricité possible dans cette seconde partie. La nature exacte du support ou du matériel nécessaire à notre jeu ne sera donc pas abordée ici afin de ne pas mêler règles de jeu et raisons techniques.

#### **II - 2.2. Perception de l'utilisateur**

##### *a) Structuration de l'information*

Notre jeu présentera une espace de jeu clair et bien découpé, dans lequel les unités des divers camps et de différents types seront clairement différenciées et où les caractéristiques des terrains seront perceptibles par un simple examen, tout en permettant le recours à un de nos mécanismes d'aide contextuelle pour plus de détails.

##### *b) Aides de jeu*

Notre jeu comprendra des aides à l'utilisateur, aussi bien en matière de mécanismes et de règles qu'en matière de contextualisation.

##### *c) Facteurs cognitifs*

Le nombre de pions à déplacer -même s'il n'a pas de limite théorique- ne devrait pas dépasser la centaine de pions afin de ne pas surcharger le joueur ; des mécanismes comme les piles ou le passage à un niveau de détail plus faible pourront être utilisés conjointement afin de réduire cette charge en conservant le réalisme de l'ensemble. Notre jeu présentera une représentation relativement classique de l'espace de jeu, avec des pions se déplaçant par exemple sur une grille

hexagonale (notre jeu se voulant générique permettra bien évidemment tous les autres modèles habituels).

## **II - 2.3. Mécanismes de jeu**

### *a) Réalisme des règles*

Les règles que nous avons élaborées et dont on pourra trouver un rapide résumé dans le point suivant, ont pour but d'assurer un certain réalisme dans la résolution des déplacements, combats et actions annexes. Bien évidemment, notre jeu permettra l'utilisation de règles différentes ou l'extension de celles-ci pour assurer un confort et un plaisir maximum à l'utilisateur par une adéquation la plus correcte possible avec ses besoins.

### *b) Corps de règles cohérent et bien pensé*

Comme tout jeu, le nôtre possède des règles de base ; nous avons voulu les garder simples tant pour des raisons de jouabilité que de clarté. En voici un rapide résumé :

#### (1) Règles générales

- Un joueur ne peut donner des ordres à ses pions que lors de son tour
- Un joueur ne voit que ce que ses unités lui permettent de voir et ce d'une façon qui peut être imparfaite
- Un pion est composé d'une ou plusieurs unités
- Un tour se termine dès que le joueur marque sa volonté de passer son tour ou après l'expiration d'un délai
- Les unités possèdent une expérience et un moral variant en fonction de leur histoire

Ces règles s'appuient sur une série de mécanismes de jeu, éminemment plus variables, dont voici les principaux :

#### (2) Règles de mouvement :

- Une unité peut se déplacer sur un ou plusieurs types de terrain.
- Lors d'un mouvement, un test est effectué afin de vérifier sa régularité.
- Si le mouvement est légal, il est effectué ; sinon pas.

- Pour être légal, un mouvement ne doit pas traverser ou viser une zone non compatible avec le type de l'unité.
- Effectuer un mouvement : après avoir déterminé le coût du mouvement projeté, on le soustrait au capital de points d'opération du pion (si ce dernier est supérieur au coût du mouvement, sinon pas de déplacement).

### (3) Règles d'attaque :

- On détermine la possibilité de l'attaque en soustrayant le coût d'une attaque au capital de points d'opération du pion.
- L'attaquant teste son moral avant de porter l'attaque. Si le test est réussi, l'attaque se poursuit, sinon, l'unité refuse de bouger (dépense de tous ses points d'opération)
- On calcule le facteur de précision en soustrayant le facteur défensif spécifique de la cible au facteur d'attaque spécifique de l'attaquant.
- Chaque unité du pion attaquant effectue des dégâts, fonction du facteur de précision, de la base de dégâts, de son expérience et d'un facteur aléatoire.
- Chaque unité du pion défenseur effectue des dégâts, fonction de son facteur de précision vis-à-vis du type d'unité attaquante, de la base de dégâts, de son expérience et d'un facteur aléatoire.
- On calcule les modificateurs au moral et à l'expérience pour défenseur comme attaquant.
- Le défenseur teste son moral. Si le test est réussi, les unités maintiennent leur cohésion ; sinon, la mobilité du pion peut être réduite ou certaines des unités s'enfuir.

### (4) Règles de défense active :

- Si un pion consacre une grande quantité de ses points d'opération à préparer sa position, il bénéficie d'un avantage si on l'attaque au cours de ce tour : il effectue en effet ses dégâts avant l'attaquant mais doit toujours tester seul son moral à la fin de l'attaque.

### (5) Règles de transport :

- Un pion peut en transporter un autre s'il possède la capacité de transporter ce type d'unité et dispose encore de suffisamment de place.
- Le pion transporté est inactif durant ce transport.

- Pour embarquer ou débarquer un pion entre un pion de transport et une localisation, il faut que les deux pions se trouvent sur des zones contiguës ou identiques.

#### *c) Options étendues*

Notre jeu a pour principale caractéristique la grande facilité de modification de ses règles, et ce de deux manières : par une simple modification des valeurs numériques des caractéristiques des unités ou des cases, permettant la simulation de situation très diversifiées ; par la modification des mécanismes de jeu en eux-mêmes en leur substituant en tout ou en partie un nouveau jeu de règles.

Dans le même ordre d'idées, notre jeu fournira divers mécanismes de découpe du temps afin de correspondre au mieux aux besoins et désirs des utilisateurs ; on peut ainsi citer, de manière non exhaustive, les mécanismes de tour par tour, de jeu simultané ou de temps réel.

Notre jeu permettra également à un nombre quelconque de joueurs de s'affronter -compte tenu de limitations matérielles inhérentes au support choisi- ; les jeux de règles et mécanismes devront bien évidemment être adaptés au nombre de joueurs afin de préserver intérêt et jouabilité.

#### *d) Multiplicité des stratégies*

Notre jeu permettra une très grande diversité de stratégies gagnantes sans permettre l'établissement de stratégies d'invulnérabilité, grâce notamment à la multitude de facteurs pris en compte et à sa grande souplesse.

#### *e) Diversité des facteurs de jeu*

Les facteurs de jeu sont en nombre théoriquement illimités dans notre jeu puisque les modifications et les surcharges des règles permettent l'introduction de nouveaux facteurs à l'infini. Parmi les mécanismes de base, citons néanmoins les actions diplomatiques entre joueurs permettant de créer des alliances d'intérêt passagères ou durables.

#### *f) Particularisation des objectifs de victoire*

Les objectifs de victoire, aisément extensibles eux aussi, pourront être particularisés avec une grande finesse, depuis la perte d'unités à la conquête de territoire en passant par l'établissement d'objectifs scénarisés, une gestion du temps ou l'atteinte de relations diplomatiques données.

## **II - 2.4. Transmission de l'information**

### *a) Richesse des mécanismes*

Notre jeu permettra d'établir une distance entre réalité et représentation à l'aide de mécanismes de filtres d'une grande souplesse ; ces filtres nous permettront ainsi de simuler avec réalisme tant les délais que la transmission incorrecte -ou volontairement erronée- d'information.

### *b) Particularisation des transmissions d'information*

Le mécanisme de filtres évoqué ci-dessus nous permettra de modifier les règles et mécanismes de transmission avec une grande finesse, par l'ajout, la modification ou la suppression de certains filtres.

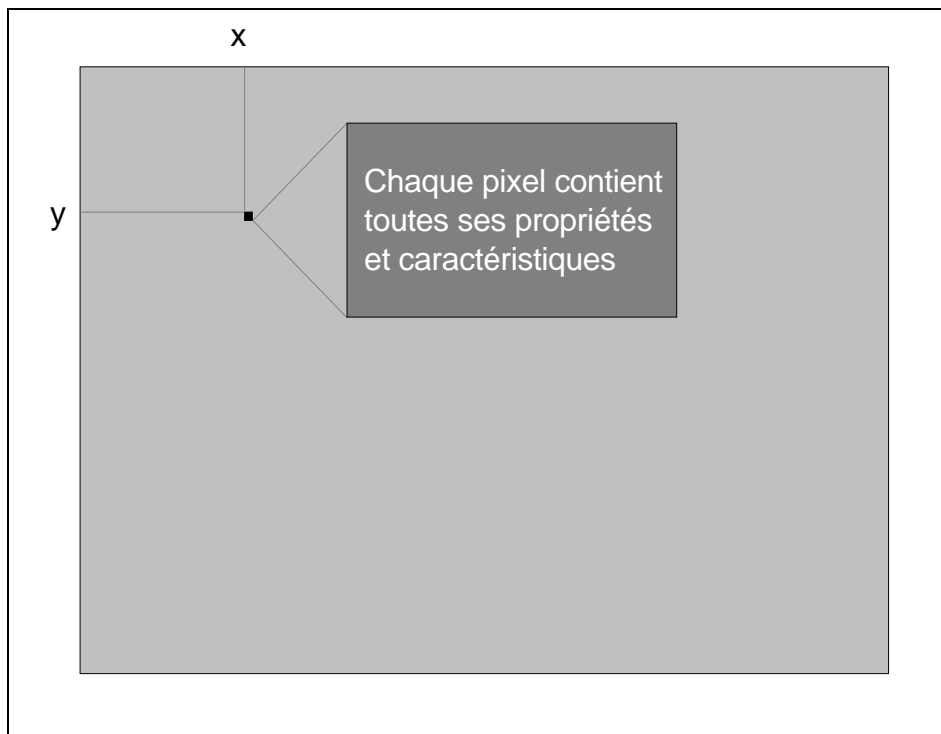
## **II - 3. Concepts**

### **II - 3.1. Perception de l'utilisateur**

#### *a) Représentation de l'espace de jeu*

Le concept de tout jeu de stratégie est basé sur le déplacement et l'interaction de « pions » localisés sur une aire de jeu, représentée par une carte constituée de différents types de terrains différant par leurs propriétés et caractéristiques. Une représentation (extrême) consisterait en un positionnement des pions uniquement en termes de coordonnées (x,y) sur la carte (deux pions physiques ne pouvant occuper la même localisation) et de « régions » physiques délimitant les terrains. Les cas limites de ce genre de représentation seraient d'une part la définition du type de terrain et des propriétés de chaque pixel (ce qui correspond déjà à une forme de discrétisation) ; d'autre part la description des régions homogènes de façon purement géométrique, par exemple polygonale, les propriétés quantitatives à l'intérieur de chaque zone pouvant alors être représentées de manière similaire, par exemple au moyen de courbes de « niveau ».





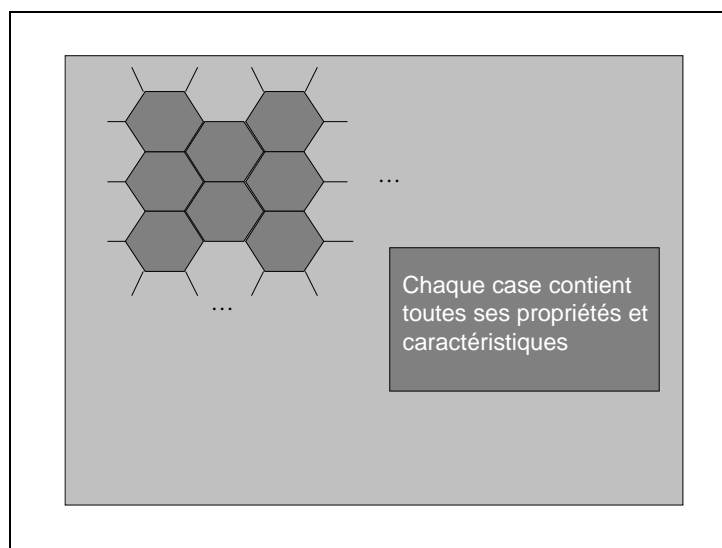
**Figure 2 : propriétés de terrain pixel par pixel**



**Figures 3 : propriétés de terrain par zones homogènes**

Une autre approche, plus classique et diamétralement opposée, consiste à subdiviser l'aire de jeu en « cases » unitaires, de géométrie régulière (souvent hexagonale dans le cas des wargames) ; les pions se déplacent alors de case en case et leur position est connue par une référence à la case qu'ils occupent (par convention, la position géographique peut être localisée au centre de chaque case,

mais sans réelle signification). Dans la mesure où les cases ont une étendue non négligeable, apparaît le problème de la présence (possible ou non) de plusieurs pions au sein d'une même case ; en fonction des mécanismes de jeu choisis, on peut exclure cette possibilité, l'autoriser pour des pions d'un même rôle, l'autoriser dans tous les cas, etc. Evidemment, la structure de données à adopter et la représentation graphique qui en sera faite dépendront fortement de ces choix conceptuels.



**Figure 4 : système par case**

Dans notre cas, le découpage en seuls hexagones conduisant à des frontières peu régulières et naturelles, nous avons été amenés à envisager le découpage des « cellules » en « zones » géométriques (simples)<sup>5</sup>. Dans cette optique, il paraît naturel de supposer que les zones présentent toutes une homogénéité de terrain et de propriétés ; il y a une subdivision hiérarchique en cellules puis en zones permettant d'envisager un héritage par défaut des propriétés. Dans ce cas, les cases de jeu naturelles sont les zones : un pion se situe dans une et une seule zone, même si certaines notions font référence à la cellule « père » (par exemple dans la détermination des cellules adjacentes) ce qui complexifie la situation. C'est ainsi que les mouvements se font de zone en zone tandis que la localisation fait référence à une zone avec représentation conventionnelle au centre de celle-ci, etc. Le problème de multiplicité des pions se pose de même au niveau des zones puisque celles-ci constituent les véritables cases de jeu.

#### *b) Découpage des hexagones en zones*

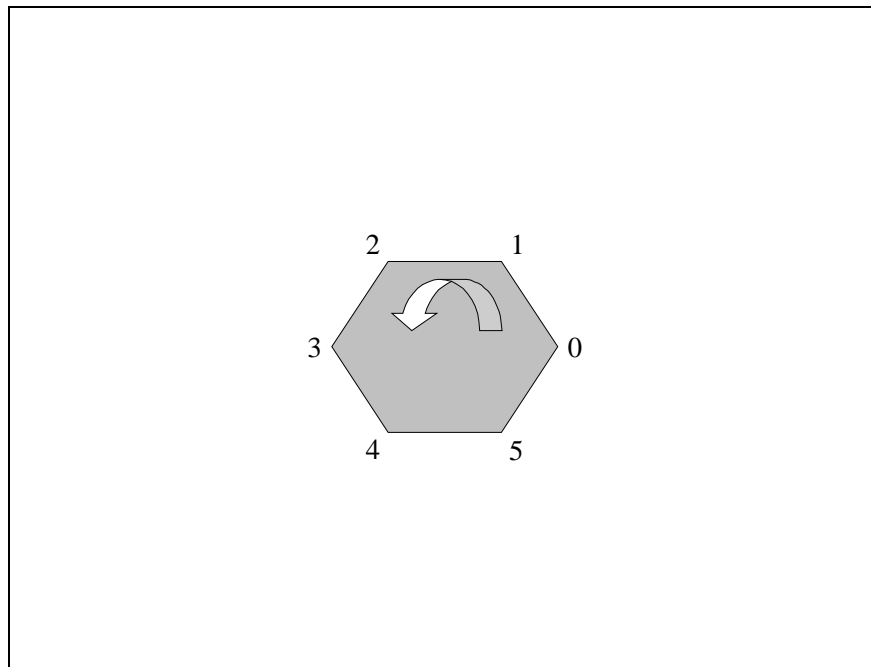
Nous avons tenté plusieurs autres découpages possibles : par les sommets, par les milieux des côtés et par des régions polygonales indépendantes des hexagones. Les essais sur cette dernière méthode donnent des performances désastreuses, du moins en utilisant les primitives existant en Java, si bien que cette approche pourtant

<sup>5</sup> Voir point suivant sur le découpage des hexagones.

séduisante a priori dut être abandonnée et nécessiterait une étude algorithmique détaillée. Dans le cadre de notre découpe, on aurait aussi pu considérer des sous-polygones élémentaires, approche éventuellement justifiée dans le cas d'une grande proportion d'hexagones subdivisés mais nécessitant un grand nombre de zones pour obtenir la même finesse...

### (1) Numérotation des sommets

Le sommet de référence est numéroté 0 et correspond au sommet « droit » de l'hexagone ; les autres sommets sont numérotés de 1 à 5 à partir de ce dernier dans le sens contraire des aiguilles d'une montre.



**Figure 5 : numérotation des sommets d'un hexagone**

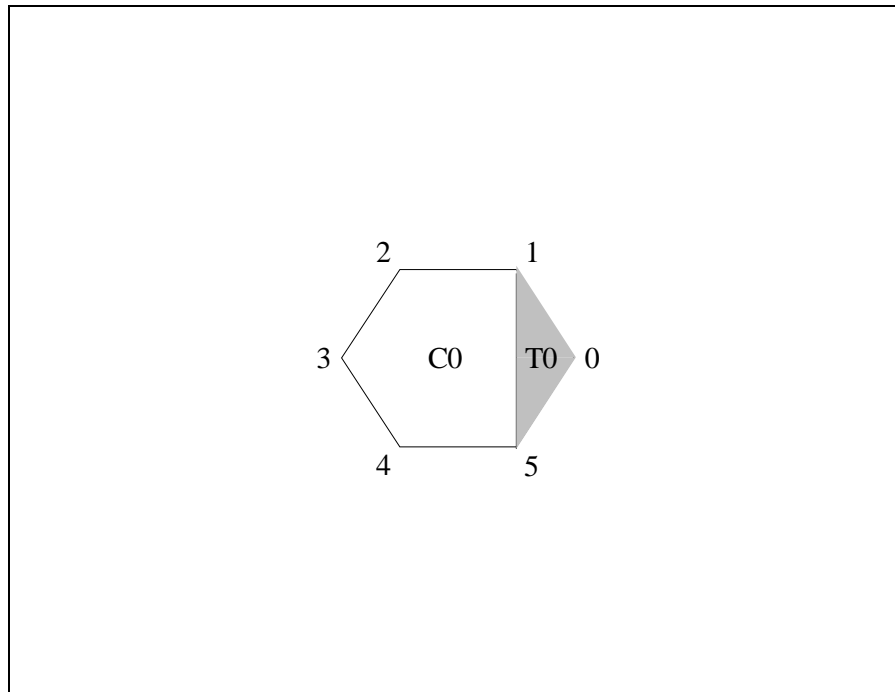
### (2) Convention de nommage des zones simples

Après découpe de l'hexagone, chaque zone est un « sous-polygone », par exemple [1,3,4,5] pour le polygone construit sur les sommets 1, 3, 4 et 5. Cette notation, qui sert à construire les polygones au sein du programme, est cependant peu commode et peu propice aux combinaisons. Afin d'identifier les zones par un nommage plus intuitif, nous avons défini les catégories suivantes :

F = F0 = FULL : hexagone complet ;

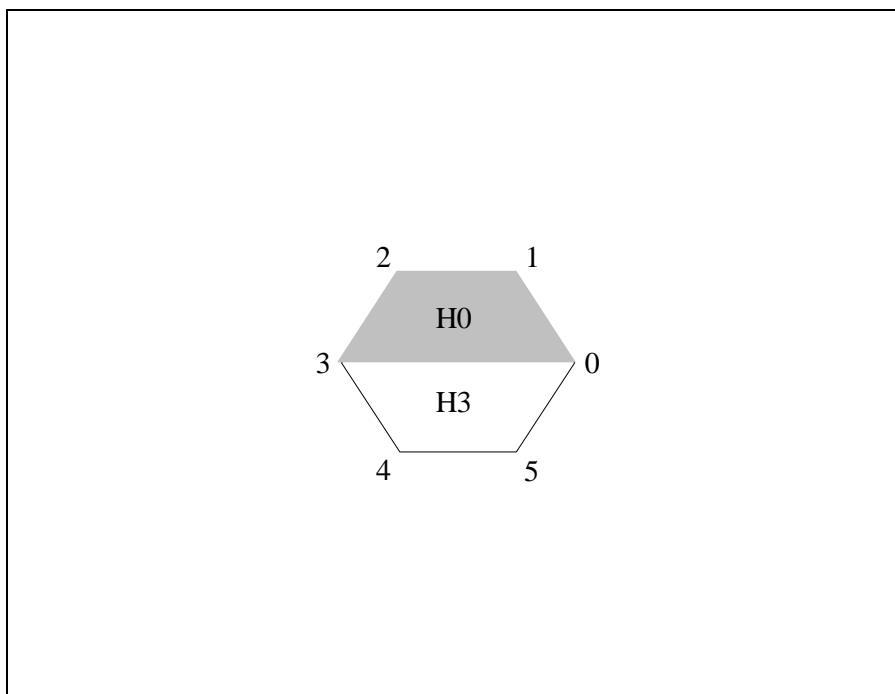
T0 (« triangle »): triangle en 0 ;

C0 (« complement »): complémentaire en 0 ;



**Figure 6 : découpe T0-C0**

H0 (« half »): divisé en deux, zone située à droite de 0 ;  
 Par symétrie, H3 est la zone de type H située à gauche de 0 ;

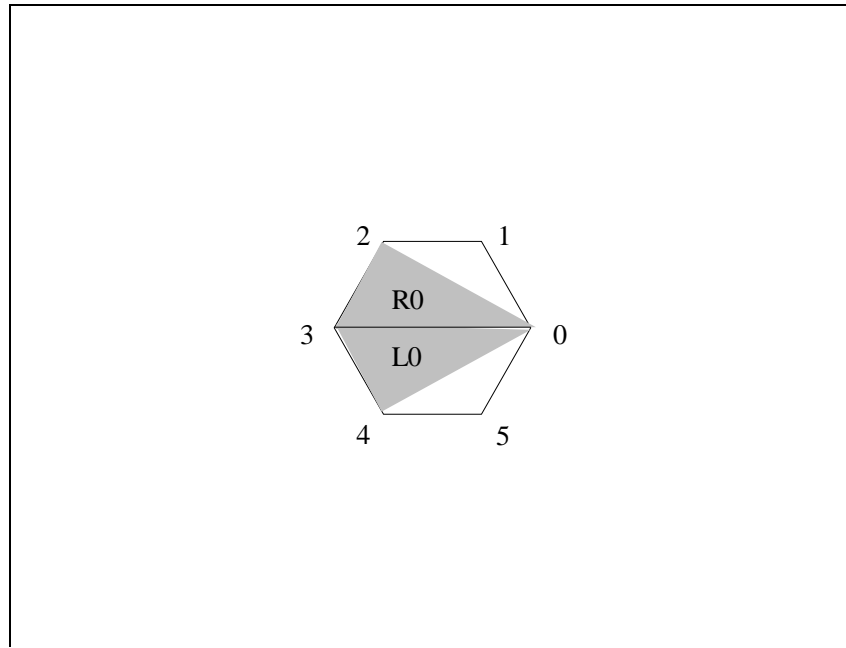


**Figure 7 : découpe H0-H3**

R0 (« right »): triangle interne à droite de 0 ;

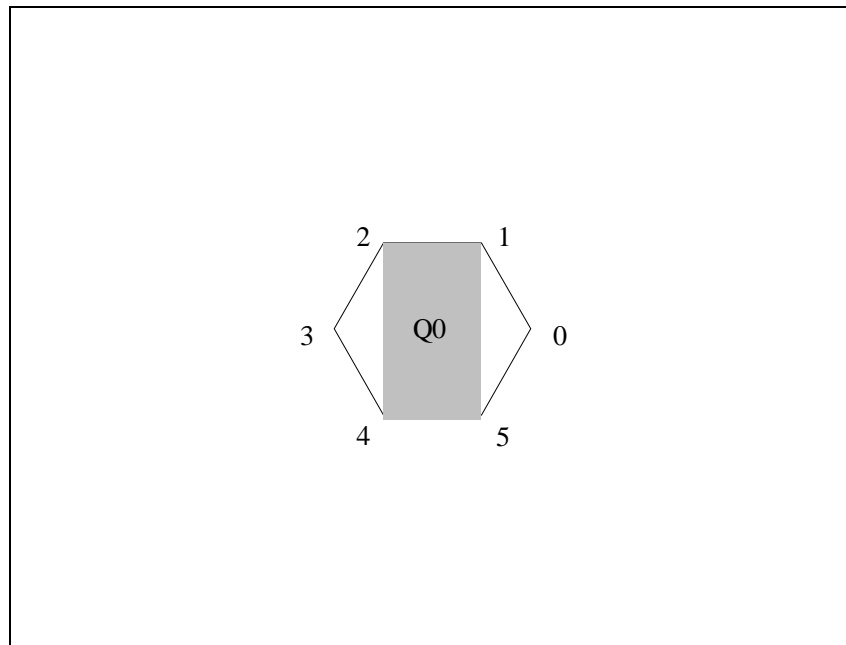
L0 (« left »): symétrique a gauche de 0 ;

B0 (« both ») :  $R0 + L0$

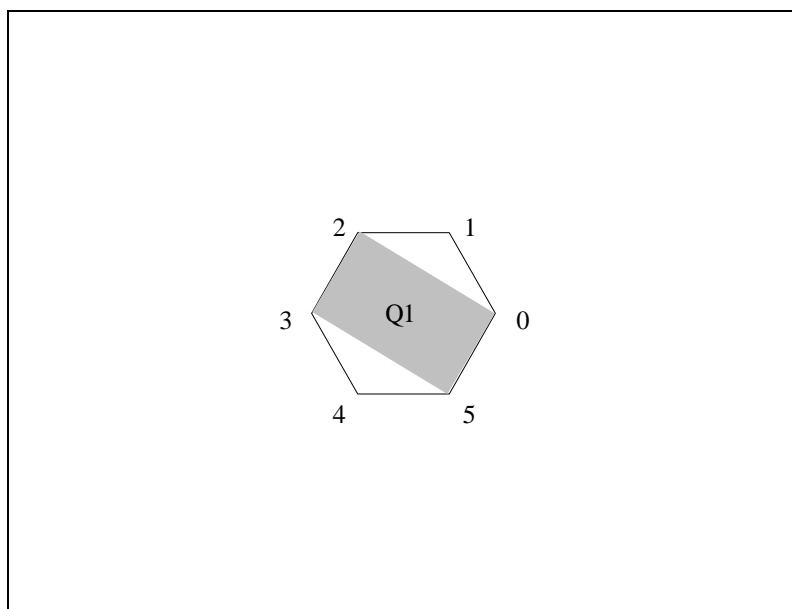


**Figure 8 : découpe R0-L0**

Q0 (« quad »): quadrilatère inscrit à l'hexagone face à 0 = hex - (T0 + T3) = (R5+R2);  
 En raison de la symétrie de la figure, Q0 est équivalent à Q3, Q1 à Q4, etc.



**Figure 9 : découpe Q0**



**Figure 10 : exemple de rotation**

Ci-dessus un exemple de rotation d'une de nos formes (Q0) pour obtenir Q1 ; de même pour les autres polygones inscrits. Chacun de ces polygones correspond donc à un vecteur d'indices définissant le polygone inscrit correspondant à partir de l'hexagone complet ; le vecteur d'indice correspondant à un polygone non-basé en 0 est obtenu par incrémentation des indices modulo 6. Ainsi, Q0 correspond au vecteur [1,2,4,5] et Q1 est obtenu via le vecteur [2,3,5,0].

### (3) Schémas de partitionnement

Ces zones simples sont utilisées pour découper l'hexagone ; de même qu'il a fallu remplacer les indices par un nommage des zones simples, il est important d'élaborer une convention de nommage des schémas de découpage en zones simples. Ceci permet à la fois d'obtenir un format de données plus simple et clair et de faciliter le dénombrement des cas possibles de partition de l'hexagone. On utilise une première forme pour démarrer le découpage, ce qui fixe l'orientation et on complète le nommage par une chaîne de caractères qui suggère les autres formes utilisées ; ainsi, T1C correspondra au découpage en T1 + C1 (voir tableau 1). On peut donc se contenter des formules de subdivision correspondant à l'indice 0, toutes les autres pouvant être obtenus par simple rotation.

**Tableau 1 : Découpes des hexagones**

Schéma de découpage	Décomposition	Nombre de cas (rotation)
F*	F0	1
C0T = T0C	T0+C0	6
Q0T = T0Q	T0+Q0+T3	3

H0H	H0+H3	3
H0L	H0+L0+T5	3
H0R	H0+R3+T4	3
B0T	B0+T1+T5	6
R0T	R0+T1+R3+T4	3
L0T	L0+T2+L3+T5	3
L0R	L0+R0+T1+T5	6

#### (4) Autres géométries

D'autres géométries peuvent bien entendu être décrites de la même manière : ainsi, dans le cas (plus simple) de cellules rectangulaires, les sommets du rectangle 0 à 3. Il n'y a dans ce cas (à part F\*) que quatre zones du seul type T : T0, T1, T2 et T3 et deux schémas autres que F\* : T0+T2 et T1+T3, soit T0T et T1T selon le même formalisme de représentation que dans le cas des hexagones. Les triangles inscrits au rectangle et les schémas de découpage correspondants pourraient aussi être décrits, par exemple, à l'aide des conventions NW (Nord-Ouest), NE (Nord-Est), SW (Sud-Ouest) et SE (Sud-Est).

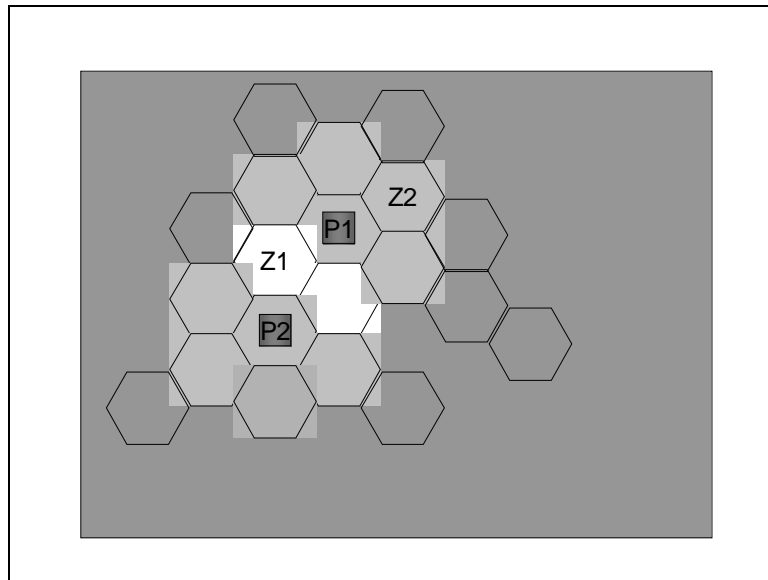
#### *c) Filtrage et vue*

Un concept qui nous semble souvent peu développé dans les jeux de stratégie est la déformation de l'information, et ce, en raison de causes multiples. Ainsi, un général éloigné du champ de bataille pourrait en avoir une vision faussée, soit qu'elle résulte d'un décalage dans le temps, soit d'erreurs de transmission ou encore de renseignements sciemment inexacts. Ce qui est vrai de ses propres troupes l'est a fortiori des troupes adverses, pour lesquelles les renseignements ne peuvent provenir que d'observations (soit par les unités amies, soit par des mécanismes complémentaires d'espions, d'éclaireurs, etc). De même, l'ennemi pourrait s'ingénier à brouiller les cartes, multipliant les unités factices ou tentant d'intercepter les messages devant avertir l'ennemi du mouvement de ses unités.

Inexactitudes, stratagèmes et délais de transmission pourraient tous être aisément considérés dans un jeu comme le nôtre : d'après nous, le moyen le plus simple de le modéliser consisterait en une série de « filtres » agissant sur les transferts d'informations et ce, à tous niveaux. Ce mécanisme permettrait de simuler une part importante de l'art de la guerre dans la plupart des conflits : l'éloignement des troupes et, partant, l'inexactitude et la non-exhaustivité des informations servant à l'établissement des plans de part et d'autre. Sur un plan purement ludique, ces mécanismes permettraient d'éviter un trop grand déterminisme de l'action, et d'augmenter la part du facteur « chance » ou encore « intuitivité ».

De plus, il n'y a guère de raison que ces mécanismes ne jouent pas sur la transmission des ordres du général vers les unités : une ordre pourrait avoir été

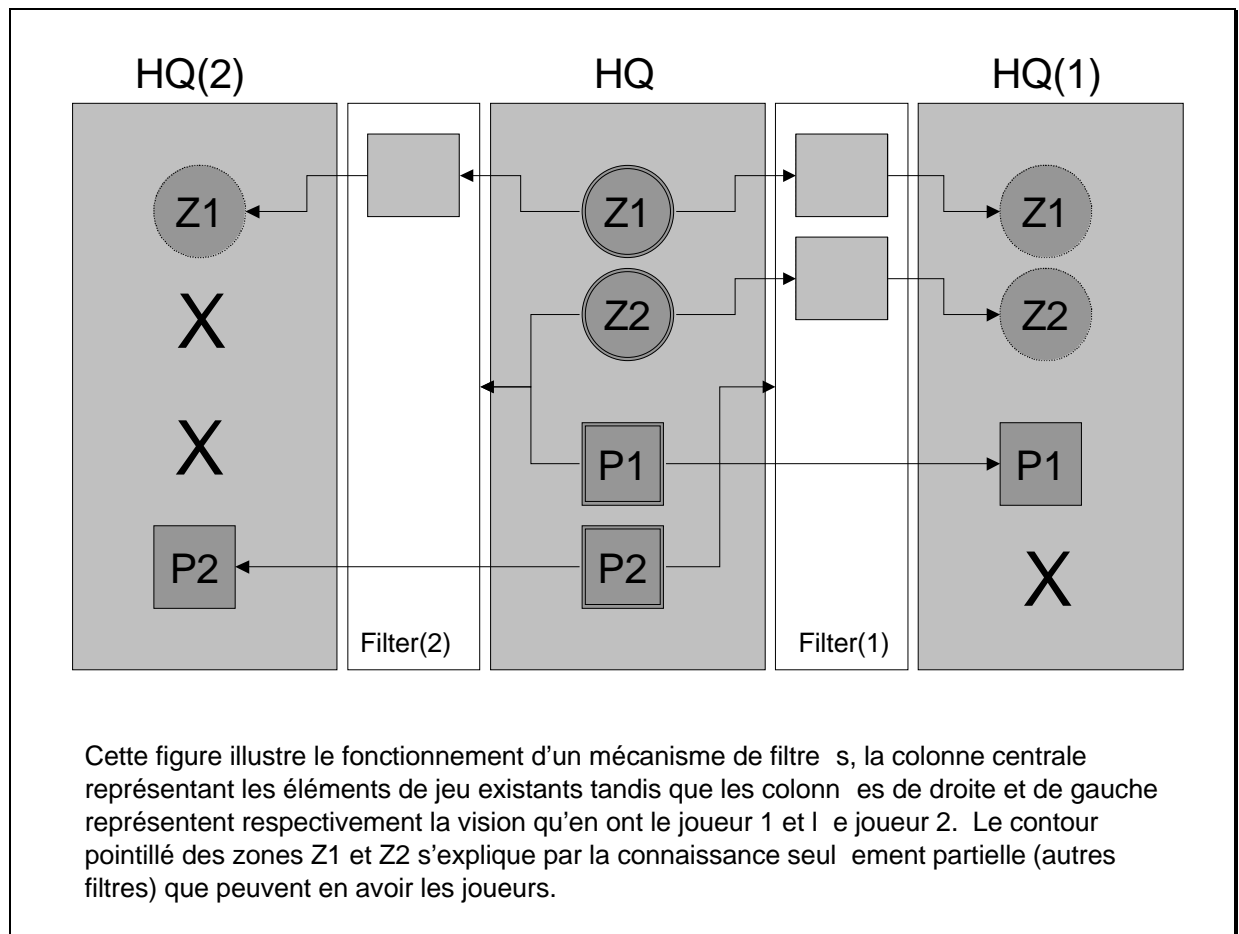
intercepté, ou retardé ; de faux ordres pourraient avoir été émis, etc. On peut le voir, ce simple concept de déformation de l'information enrichit considérablement le jeu et permet de moduler, grâce à l'importance des déformations, la facilité et la rapidité de transfert de l'information pour simuler au mieux une situation donnée ou simplement fournir une situation de jeu intéressante.



**Figure 11 : état de jeu de l'exemple**

Afin de mieux expliciter notre vision la plus basique à cet égard, prenons l'exemple d'un jeu de stratégie où chaque joueur connaît tous ses pions mais aussi toutes les structures éloignées au plus d'une case d'un de ses pions. Le joueur 1 connaît donc le sous-ensemble des informations accessibles au joueur pour son pion P1, et les informations autorisées au joueur pour la zone 1 et la zone 2. Le joueur 2, outre son pion P2, ne connaît que la zone Z1. De façon intéressante, chaque joueur ignore la présence d'une unité ennemie à proximité en raison de la faible visibilité au sein de ce jeu.





**Figure 12 : mécanismes de filtre et vision résultante**

Des jeux existants extrêmement connus employant un mécanisme à jeu caché sont légion : bataille navale, stratège, etc. Par ailleurs, ce mécanisme permet de refléter dans un jeu de stratégie plus réaliste le caractère incertain de ces transmissions et le délai parfois non négligeable d'acheminement de l'information. Un des meilleurs exemples historiques en est le rôle joué par les estafettes ou leur équivalent durant une grande partie de l'histoire militaire et l'importance cruciale qu'une dépêche interceptée pouvait revêtir dans le déroulement d'une bataille, d'une campagne ou encore d'un conflit.

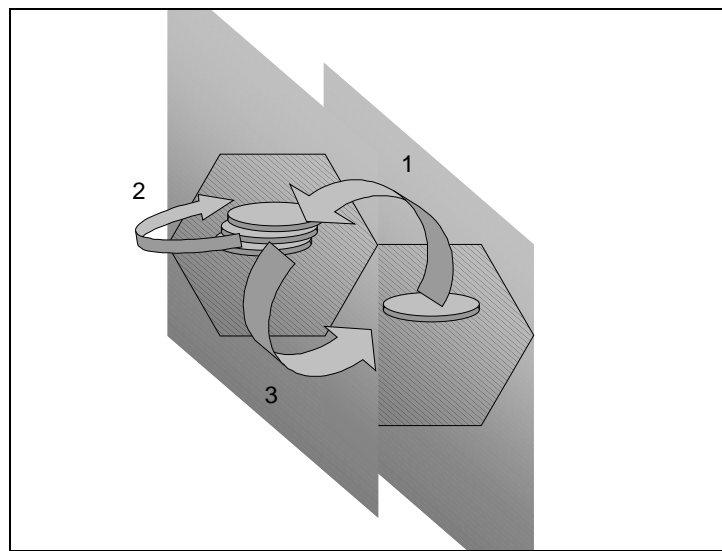
## **II - 3.2. Mécanismes de jeu**

### *a) Multi-pions (piles)*

Le premier problème posé par l'autorisation de placer plusieurs pions dans une même zone de jeu consiste à rendre perceptibles à l'utilisateur additions, retraits et mouvements. Souvent, les concepteurs de jeux de stratégie ont recouru à la simple superposition des pions ; plus rarement, un nouveau pion remplace les anciens afin de rendre de manière plus commode leur combinaison. En tous les cas, cette représentation est problématique de par la nature complexe des unités : ainsi, comment rendre graphiquement la présence en une zone d'une flotte transportant

par ailleurs des centaines de chasseurs, des troupes d'invasion et des engins amphibies ?

L'autre problème causé par l'existence de telles structures est d'imaginer un système de combat « valable » tant au niveau de sa résolution en elle-même que de ses conséquences ; de multiples solutions ont déjà été proposées dans divers jeux de stratégie, allant de l'impossibilité d'une attaque par pile (chaque pion de la pile devant recevoir un ordre d'attaque propre contre un adversaire spécifique, lui-même éventuellement membre d'une autre pile), en passant par des formes d'automatisation de ce processus par la résolution successive des attaques de chaque pion de la pile contre un adversaire donné, leur effet étant cumulatif, à l'intégration des caractéristiques des unités en un « bloc » unique pouvant ensuite se prêter à une résolution du combat unique.



**Figure 13 : gestion d'une pile de pions**

Les trois problèmes essentiels d'une gestion de pile -sans prendre en compte la mobilité de la pile elle-même- sont l'ajout d'un nouveau pion (1), la sélection d'un pion parmi ceux de la pile (2) et le retrait d'un pion de la pile. Les deux approches essentielles -conglomérat ou simple superposition spatiale- nous semblent bien illustrées par les exemples du jeu de Risk (résolution des batailles par catégories quantitatives -1 pion, 2 pions, 3 pions et plus-) pour l'aspect de combinaison des pions en méta-pions et de Civilisation (résolution des conflits pion par pion malgré la possibilité de créer des piles de pions) pour l'aspect superposition spatiale.

Ce même mécanisme de pile peut donc représenter des concepts fort différents, de la simple juxtaposition spatiale de pions, parfois regroupés dans un objet unique pour des raisons de commodité, à la création de « meta-pions » intégrant les diverses caractéristiques de leurs pions constitutifs.

## *b) Propriétés des pions*

**Tableau 2 : Propriétés des pions**

Propriété	Echelle de valeur dans notre exemple
Moral	0-100
Expérience	0-n
Valeurs offensives	0-5
Valeurs défensives	0-5
Facteur de dégâts	0-1000
Capacité de coups/unité	0-10000
Nombre maximum d'unités	1-50
Capacité de transport	0-500
Points d'opération	0-50

### (1) Moral

Le moral représente la volonté et la combativité des troupes constituant le pion. Il est testé pour l'attaquant avant d'évaluer les conséquences d'une attaque, pour le défenseur après cette évaluation. Ce facteur peut entraîner des conséquences importantes pouvant aller jusqu'à la dispersion des troupes.

### (2) Expérience

A mesure qu'elles combattent, les troupes deviennent plus expérimentées, tirant avec plus de précision et causant dès lors plus de dommages. L'expérience est un des facteurs entrant en compte dans le calcul des dégâts occasionnés.

### (3) Valeurs offensives

Certaines unités sont plus puissantes que d'autres, mais ne montrent pas les mêmes capacités contre différents ennemis. Chaque Pion possède un facteur d'attaque contre les unités terrestres, maritimes et aériennes.

### (4) Valeurs défensives

Des unités différentes seront plus ou moins endommagées selon la nature de l'attaque ; ainsi, une unité peut se révéler très bien défendue contre un type d'attaquant mais vulnérable à un autre.

#### (5) Facteur de dégâts

Un type d'unité possédant une grande puissance de feu causera plus de dommages.

#### (6) Capacité de coups

La capacité de coups d'une unité représente une mesure de la quantité de dégâts qu'une unité peut subir avant d'être détruite.

#### (7) Nombre maximum d'unités

Le nombre maximum d'unités est propre à chaque type d'unité ; généralement, plus l'unité est grande et complexe, moins un pion en comportera. Ce facteur est essentiellement dû à une volonté de faciliter le processus cognitif des joueurs.

#### (8) Capacité de transport

La capacité de transport représente le nombre d'unités qu'un pion donné peut transporter.

#### (9) Points d'opération

Les points d'opération sont une mesure de la mobilité de l'unité.

### *c) Propriétés des hexagones*

**Tableau 3 : Propriétés des hexagones**

Propriété	Echelle de valeur dans notre exemple
Relief	0-6000
Difficulté	0-100
Découpage	Spécial

Il nous apparaît essentiel de bien marquer la différence entre les zones et les hexagones : les zones sont de nature uniforme, les éléments atomiques de subdivision des hexagones ; les hexagones correspondent à un niveau supérieur de découpe conservés pour des raisons essentiellement historiques (la majorité des

jeux de stratégie existants y ayant recours). Dans un souci de simplification, mais aussi en raison de nos choix lors du développement de notre jeu, les propriétés « relief » et « difficulté » sont propres à chaque hexagone et donc identiques pour toutes les zones de celui-ci.

### (1) Relief

Le relief est un concept équivalant à l'altitude (pour la terre) ou la profondeur (pour la mer), rendant le déplacement plus ou moins aisé.

### (2) Difficulté

La difficulté traduit le fait que certains endroits présentent des particularités qui rendent leur traversée plus ou moins aisée.

### (3) Découpage

Au sein de chaque hexagone, on peut trouver différents types de terrain (terre et mer dans notre exemple) en fonction des différents découpages possibles en zones.

## ***II - 4. Conclusion***

Dans cette deuxième partie, nous avons pu définir notre jeu, ses mécanismes et règles (de façon générale) ; nous avons aussi dû poser un certain nombre de choix restrictifs durant ce processus de définition. Notre jeu est maintenant prêt à être transposé sur un support quel qu'il soit.

## **III.3<sup>ème</sup> partie : Informatisation**

### ***III - 1. Introduction***

Le développement de l'informatique au cours des dernières décennies a, non seulement, sans cesse accru les champs d'application de ces techniques, mais aussi les moyens disponibles, rendant possibles des jeux de plus en plus ambitieux, tant dans leurs aspects techniques que sur le plan de leur complexité. Malheureusement, cette course à la performance s'est parfois effectuée au détriment du contenu et de l'intérêt ludique de certains jeux ; cette évolution a d'ailleurs été encouragée par les méthodes d'évaluation employées et les grilles de critères retenues par une partie de la presse spécialisée<sup>6</sup>.

Le terme de « jeu de stratégie » recouvre aujourd'hui une catégorie de jeux très vaste aux buts et systèmes multiples, tant en terme de mécanismes (tour par tour, ordres simultanés, temps réel), de représentations (vue en plan, vue en trois dimensions, vue subjective, etc), de système de communication (« e-mail game », applet/servlet, application, etc) ou de style de jeu (gestion, conquête ou encore « D&C » -develop and conquer-), qu'en matière de réalisme (« reconstitutions », simulations, simple vernis stratégique, etc).

Dans un premier temps, nous tenterons de poser le problème en termes informatiques, de définir les contraintes imposées par nos choix conceptuels. Dans un second temps, nous expliquerons nos choix, leurs avantages et inconvénients éventuels. Nous ne pourrons malheureusement pas aborder la question des standards concernant les différents domaines évoqués en raison de l'ampleur de la tâche.

### ***III - 2. Concepts informatiques***

#### **III - 2.1. Objets principaux**

##### *a) Joueurs*

Les joueurs sont les participants humains à une session : ils sont identifiés par un nom de joueur et un mot de passe (et rien d'autre, les autres informations consistant en références vers d'autres objets de jeu). Lors de leur connexion au serveur, celui-ci vérifie la concordance des informations fournies avec celles contenues dans un

---

<sup>6</sup> Ainsi, les magazines d'informatique ludique -ex : Joystick- consacrent en moyenne entre la moitié et les deux-tiers de leurs articles à l'analyse des aspects visuels et d'immersion des joueurs au détriment d'aspects plus arides mais néanmoins fondamentaux comme les mécanismes de jeu ou l'univers de référence.

fichier ; si c'est le cas, il attribue un rôle (le premier disponible) au joueur. Chaque joueur, par le biais du mécanisme de rôle, ne peut commander que les unités associées à son quartier général (HQ) et ce uniquement lors de son tour.

#### *b) Rôles*

Il s'agit des « emplacements » destinés à accueillir les joueurs : chacun est caractérisé par un identifiant et peut présenter des spécificités par rapport aux autres, tant au niveau des caractéristiques des unités que de ses liens avec d'autres rôles, etc.

#### *c) Client*

L'application client gère l'ensemble des fonctionnalités de représentation et de concordance avec le serveur. Il peut y avoir plusieurs applications clients connectées sur un même serveur ; a priori, il peut de même avoir plusieurs joueurs connectés à un même client.

#### *d) Serveur*

L'application serveur du jeu : elle gère les interactions entre rôles, met à jour les données des unités et répercute ces modifications vers les différents rôles concernés par le biais d'un mécanisme de filtres.

#### *e) Session*

La « partie » de jeu à proprement parler : dans un contexte unique (ressources chargées en début de session), elle voit les différents joueurs interagir avec l'environnement de jeu. Un scénario est une séquence « encapsulée » et prête à la réutilisation de lignes de commandes.

#### *f) Cellule*

La cellule, dans notre cas un hexagone, est un des éléments de la grille de découpe de la zone de jeu. La cellule peut adopter des formes très diverses (maillage) ; dans notre exemple, elle est découpée en zones.

#### *g) Pion*

Comme le nom l'indique, il s'agit d'un pion, dont les instanciations sont la représentation des différentes unités en jeu.

#### *h) Pion Factory*

Cet objet prend en charge la création des pions des différents protagonistes. Il est également utilisé dans les mécanismes d'arbitrage et de gestion d'une partie.

*i) Terrain*

Il s'agit des différentes catégories de sols pouvant constituer une zone d'un hexagone. Ces catégories sont vérifiées lors des déplacements, chaque unité ayant des terrains spécifiques et/ou interdits.

*j) Zone*

Subdivision d'un hexagone selon un ou plusieurs segments joignant des sommets de l'hexagone.

### **III - 2.2. Objets secondaires (regroupement)**

*a) Grille*

La grille est le maillage de cellules couvrant l'espace de jeu : afin de lui assurer une structure régulière, il convient de choisir des polygones réguliers (rectangles, triangles, hexagones, etc).

*b) Quartier général*

Le Quartier général (ou HQ) est le centre névralgique de chacun des camps en présence ; c'est là que sont regroupées les informations relatives à la géographie et aux unités et de là que sont issus les ordres. Tous ces quartiers généraux dépendent d'un quartier général principal, regroupement de toutes les informations concernant l'état du jeu.

*c) Champ de bataille (espace de jeu)*

Il s'agit de la représentation des informations sur l'univers de référence : aspects géographiques et unités tels que connus par le quartier général du rôle concerné y sont intégralement repris.

### **III - 2.3. Résumé architectural**

Serveur (logique, 1/port) → « moteur » de jeu
Session (1/serveur) = partie
Client    application se connectant a un serveur
Joueur (1 à n)    connecté via un client à un serveur ; assigné à un rôle dans une session
Rôle (1 à n)    « camp » en présence au sein d'une session



Chacun de ces éléments est logiquement dépendant des autres, comme encapsulé.

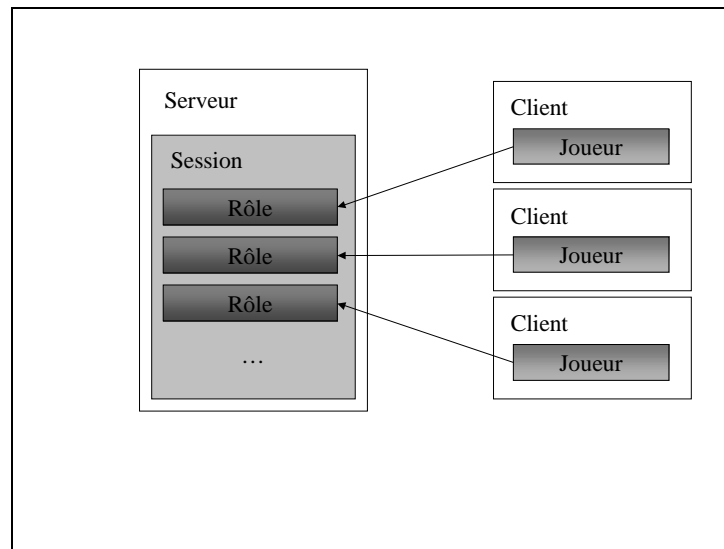


Figure 14 : architecture générale

### **III - 3. Questions d'informatisation**

#### **III - 3.1. Structuration générale**

1) Il peut y avoir plusieurs serveurs (ports) dans notre logiciel, sur une même machine virtuelle ; cela nous permet donc de lancer n serveurs sur n ports différents (aux ressources matérielles près).

2) A priori, les sessions sont bien sûr indépendantes des serveurs : un même joueur peut choisir de se connecter au serveur de son choix en spécifiant l'adresse de celui-ci ; par extension, il peut y avoir plusieurs sessions par serveur sans interaction entre elles. Le joueur est amené à choisir la session qu'il souhaite rejoindre lors de sa connexion.

Nous avons décidé d'appliquer l'hypothèse simplificatrice d'une seule session par serveur : en effet, si nous voulions offrir de plus vastes possibilités en la matière, il serait plus logique de recourir à un cadre plus riche en facilités multijoueurs, comme un serveur http, des servlets et des sessions (http s'entend).

3) A priori, un joueur dans une session peut être associé à plusieurs rôles : à partir d'un même identifiant, le joueur peut gérer successivement (même au sein d'une même session) plusieurs rôles au cours d'un tour de jeu.

Par souci de simplification, lorsque le joueur se connecte, on peut lui affecter automatiquement un rôle prédéfini, par exemple, le premier encore non affecté correspondant à ce joueur. Dans le cadre de notre hypothèse, on affecte un et un seul rôle par joueur.

4) Pour réduire le temps de développement, nous avons choisi de ne pas gérer

reprises, sauvegardes, etc., tant au niveau du serveur qu'au niveau du client : il y a donc une reconstruction systématique et totale par le serveur à chaque session ; de même, le client décharge l'intégralité des données nécessaires au début de chaque connexion.

### III - 3.2. Choix d'implémentation

Quelques questions viennent immédiatement à l'esprit :

#### *a) Choix du/des langage(s)*

Quels langages choisir ; faut-il un langage serveur et un langage client, des assemblages de langages tels ceux entre un langage de script interprété comme Perl, TCL, Python (fournissant la structure et la logique d'ensemble) et un langage orienté objet et compilé comme C++ (pour les fonctions de base) ou tout développer sous le même langage (par exemple Java)?

**Nous avons choisi de développer, aussi bien pour le serveur que pour le client, en utilisant un seul langage : dans cette optique, nous avons retenu Java.**

En effet, il est plus facile d'obtenir du code sans erreur en Java qu'en C++ (un code C++ contiendrait en moyenne une erreur toutes les cinquante lignes) ; pour ce faire, Java dispose d'un mécanisme d'exception contraignant dans lequel une exception ne peut pas être ignorée. Une méthode appelant une méthode pouvant provoquer la levée d'une exception doit soit traiter ce type d'exception, soit indiquer explicitement qu'elle ne la traite pas et fait remonter ce type d'exception. Une exception non captée par aucune méthode provoque l'arrêt de la JVM.

Il n'y a pas d'allocation ou de libération de mémoire manuelle en java ; cet aspect est géré de façon purement automatique.

L'arithmétique des pointeurs présente en C++ a été remplacée par un véritable tableau d'objets, éliminant du même coup tous les problèmes liés à un compteur erroné.

L'héritage multiple a été remplacé par la notion d'interface.

Application comme applet sont possibles et réalisables, grâce à deux classes loader spécifiques par exemple.

On peut employer le même langage de programmation pour les clients comme pour le serveur (simplification de la maintenance, notamment évolutive)

Disponibilités d'outils de développement (development toolbox) et de documentation (javadoc).

Deux exemples réussis nous ont confortés dans ce choix : d'une part, robocode (IBM alphaworks), une application mettant en scène une animation en temps réel de

plusieurs robots (chaque robot étant géré par son propre « thread ») ; d'autre part, barisko (ou JRisk), implémentation en java du célèbre jeu Risk.

Le mieux est peut-être de reprendre les caractéristiques de java telles que décrites par ses concepteurs dans leur livre blanc<sup>7</sup> :

Distribué : il existe une grande variété de bibliothèques permettant de gérer les protocoles TCP/IP tels que http et ftp en java (comme en Perl) mais intégrées et aisément modifiables par le mécanisme de « surcharge ». Les applications java peuvent accéder à des données sur des systèmes distants avec autant de facilité qu'à des ressources locales grâce au mécanismes des URL et les objets distribués peuvent communiquer entre eux grâce au RMI (remote method invocation).

Fiable : Java a été conçu pour que les programmes qui l'utilisent soient fiables sous différents aspects. Sa conception encourage le programmeur à traquer préventivement les éventuels problèmes, à lancer des vérifications dynamiques en cours d'exécution et à éliminer les situations génératrices d'erreurs... La seule et unique grosse différence entre C++ et Java réside dans le fait que ce dernier intègre un modèle de pointeur qui écarte les risques d'écrasement de la mémoire et d'endommagement des données.

Orienté objet : comme les autres langages orientés objet, java se concentre sur les données (les objets) et sur les interfaces avec les objets. Une classe se compose traditionnellement d'attributs et de méthodes dont l'encapsulation est pondérée par des critères de visibilité sur 5 niveaux. Les attributs sont soit des types primitifs (boolean, char, int, float ou double) soit des références sur des objets. Toute classe hérite d'une et d'une seule classe (par défaut, de la classe mère java.lang.Object) mais peut implémenter plusieurs interfaces. Une interface est une classe où toutes les méthodes sont abstraites (i.e. définie mais non implémentée). Java supporte le polymorphisme par héritage de classes (e.g. : si C est une classe et r une variable de type C, r peut référencer toute instance de C ou des classes dérivées de C) et par implémentation d'interface (e.g. : si I est une interface et r une variable de type I, r peut référencer toute instance de classe implémentant I).

Simple : java se veut un langage permettant une programmation simple tirant parti de l'expérience standard actuelle. En conséquence, même si C++ comportait de nombreux défauts aux yeux des concepteurs, Java a été conçu de façon relativement proche de ce langage dans le dessein de faciliter la compréhension du système. De nombreuses fonctions compliquées, mal comprises, rarement utilisées de C++, qui semblaient par expérience apporter plus d'inconvénients que d'avantages, ont été supprimées de Java.

Sûr : Java a été conçu pour être exploité dans des environnements serveur et distribués. Dans ce but, la sécurité n'a pas été négligée. Java met l'accent sur la construction de systèmes inaltérables et sans virus.

---

<sup>7</sup>. Grand (M.), *Java: Language Reference*.

Architecturalement neutre : Le compilateur génère un format de fichier objet dont l'architecture est neutre – le code compilé est exécutable sur de nombreux processeurs, à partir du moment où le système d'exécution de Java est présent. Pour ce faire, le compilateur Java génère des instructions en bytecode qui n'ont de lien avec aucune architecture particulière. Au contraire, ces instructions ont été conçues pour être à la fois faciles à interpréter et faciles à traduire en code natif.

Portable : A la différence du C/C++, on ne trouve pas les aspects de dépendance de la mise en œuvre dans la spécification. Les tailles des types de données primaires sont spécifiées, ainsi que le comportement arithmétique qui leur est applicable.

Interprété : L'interpréteur Java peut exécuter les bytecode directement sur n'importe quelle machine sur laquelle il a été porté. Dans la mesure où la liaison est un processus plus incrémentiel et léger, le processus de développement peut se révéler plus rapide et exploratoire.

Performant : En général, les performances des bytecodes interprétés sont tout à fait suffisantes, il existe toutefois des situations dans lesquelles des performances plus élevées sont nécessaires. Les bytecodes peuvent être traduits à la volée en code machine pour l'unité centrale destinée à accueillir l'application. Java représente ainsi un bon compromis entre interpréteur et compilateur (C++ / Java / Perl : de moins en moins performants).

Multithread : Les avantages du multithread sont une meilleure interréactivité et un meilleur comportement en temps réel. Toute instance peut être gérée dans un nouveau thread. Pour cela, elle doit soit hériter de la classe `java.lang.Thread` ou implémenter l'interface `java.lang.Runnable`. Les accès concurrents sont gérés par des monitors transparents pour le programmeur. Le mot clé `synchronized` permet de rendre exclusif l'accès à un attribut, un bloc de code ou une méthode.

### *b) Modèle de développement*

Quelle nature/classification pour le programme : en fonction d'une représentation orthogonale entre propriétaire et open-source ; payant et freeware ?

**Nous avons choisi de faire de notre programme un « freeware open-source » donc un programme gratuit au code accessible à tous.**

En effet, notre but est non seulement de proposer un jeu gratuit et aisément modifiable par ses futurs utilisateurs pour être adapté à leurs besoins spécifiques, mais aussi d'encourager les améliorations et reprise du projet par des tiers, le modèle open-source étant par essence collaboratif.

### *c) Gestion des données*

Quel modèle de gestion des données : repository, 2 tiers/3 tiers, unifié, applet, servlet, etc ?

**Pour un jeu à taille réelle, il conviendrait de choisir un modèle de gestion des données 3 tiers ; dans notre implémentation minimale, la partie base de données n'a pas de raison d'être et nous nous avons simplement chargé les données en mémoire au départ de fichiers « properties ».**

L'architecture 3-tiers est composée de trois éléments, ou plus précisément dans ce cadre là de trois couches. En effet dans ce contexte, et dans la philosophie qui a guidé l'élaboration de cette architecture, il est plus adéquat de parler de couche fonctionnelle où à chacune d'elle est attaché un élément/entité logique<sup>8</sup>.

Ainsi, dans le modèle 3-tiers il faut distinguer trois couches/éléments :

La couche présentation (ou affichage si l'on souhaite) associée au client qui de fait est dit "léger" dans la mesure où il n'assume aucune fonction de traitement à la différence du modèle 2-tiers.

La couche fonctionnelle liée au serveur, qui dans de nombreux cas est un serveur Web muni d'extensions applicatives.

La couche de données liée au serveur de base de données (SGBD)

D'un point de vue général quelques points importants sont à souligner pour l'architecture 3-tiers :

Le client qui n'a donc que des fonctions d'affichage ne fait que des requêtes vers le serveur, aucun calcul n'est effectué par le client. Les résultats de ses requêtes sont ensuite affichées.

C'est le serveur qui va effectuer tous les calculs ou faire des requêtes vers d'autres serveurs additionnels (ie vers des SGBD).

Les avantages de l'architecture 3-tiers sont principalement au nombre de quatre :

1. Les requêtes clients vers le serveur sont d'une plus grande flexibilité que dans celles de l'architecture 2-tiers ; en effet les appels clients ne spécifient que des paramètres et des structures de données pour les valeurs de retour.
2. Faisant le pendant avec la première remarque, l'utilisateur n'est pas supposé connaître le langage d'interrogation de la base de données, qui ne sera pas implémenté dans la partie client qui ne s'occupe que de fonctions d'affichage. De fait des modifications peuvent être faites au niveau du SGBD sans que cela impacte la couche client. Par ailleurs, on peut très bien envisager une organisation des données sans présupposition quant au langage lui même et à leur organisation (relationnelle, hiérarchique, etc). Cette flexibilité permet à une entreprise d'envisager dans le cadre d'une architecture 3-tiers une grande souplesse pour l'introduction de toutes nouvelles technologies.

---

<sup>8</sup> McConnell (S.), *Code Complete*.

3. D'un point de vue développement, la séparation qui existe entre le client, le serveur et le SGBD permet une spécialisation des développeurs sur chaque tiers de l'architecture.
4. Plus de flexibilité dans l'allocation des ressources; la portabilité du tiers serveur permet d'envisager une allocation et ou modification dynamique au grés des besoins évolutifs au sein d'une entreprise.

Les inconvénients sont au nombre de deux :

1. Une expertise de développement à acquérir qui semble plus longue que dans le cadre d'une architecture 2-tiers.
2. Corollaire du point précédent, les coûts de développement et de maintenance sont au début plus élevés pour une architecture 3-tiers que pour du 2-tiers.

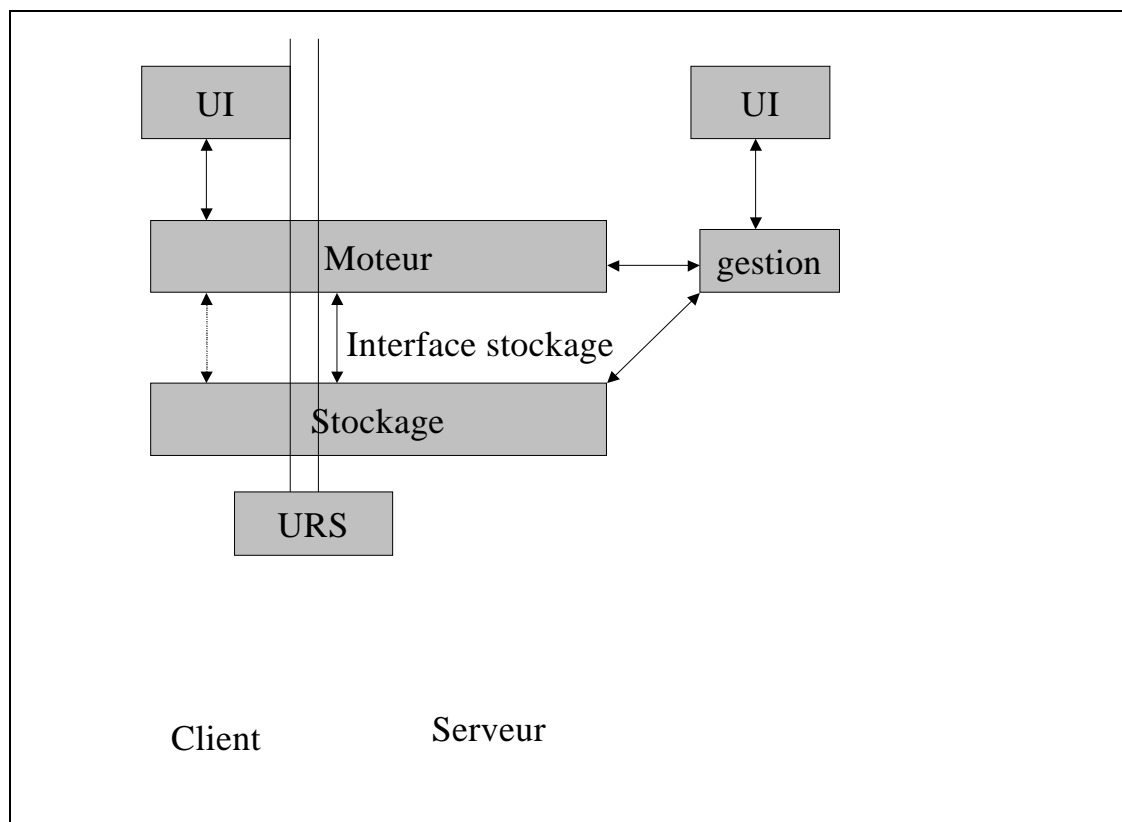


Figure 15 : architecture logique

#### d) Etats de jeu

Quels états : passer tour, blocage, etc ?

**Dans notre mise en oeuvre de démonstration, en l'absence de gestion temporelle des états de jeu, nous n'avons pas mis en oeuvre les états correspondant à un joueur déconnecté ou absent.**

Dans un jeu plus développé, le joueur pourrait être actif (en cours de tour), passif (en attente de son tour), bloqué (passe tous ses tours) ou spectateur (pas d'interaction avec le jeu).

#### *e) Objets de jeu et répartition client/serveur*

Quels objets de jeu : statiques (fond, cartes, terrains, etc), dynamiques (unités, etc), quelles interactions entre eux et comment les répartir entre client et serveur?

**Nous avons essayé de coller le plus possible au réel (par la correspondance entre objets physiques et leurs modèles informatiques)<sup>9</sup>. Tous les objets existent au niveau serveur et se trouvent répliqués au niveau client après filtrage éventuel.**

A côté de ces objets, « images » informatiques du monde réel, nous avons introduit une série d'objets plus spécialisés correspondant à des tâches exécutables (transporteur), des objets graphiques et des services de communication (infrastructure).

#### *f) Gestion temporelle des états des données*

Quels mécanismes de mises à jour : (snapshot /= replay, timestamp = last update, etc) ?

**Nous n'avons pas implémenté ce type de mécanismes.**

Dans une implémentation destinée à un serveur de production, une opération de sauvegarde est nécessaire afin de pouvoir restaurer l'état des données des parties en cours. Les informations disponibles à un moment sur le serveur et sur le client résultent des objets de jeu provenant de fichiers sources et mis à jour par les interactions du jeu. Pour le serveur, une situation à un instant donné peut, par exemple, être reconstruite au départ d'un « snapshot » et de l'histoire des interactions depuis celui-ci. Pour le client, elle s'obtient depuis le serveur ; ici encore, plusieurs stratégies de reconstruction sont possibles afin d'optimiser la communication client/serveur lors de reconnections par exemple.

Complémentairement à ces mécanismes relatifs à ces objets de jeu, le bon fonctionnement du serveur nécessite d'autres fonctions d'administration de l'infrastructure de jeu : gestion d'utilisateurs, de leurs droits d'accès, de participation, etc.

Par ailleurs, il importe de classer logiquement et efficacement les données employées dans le cadre du jeu afin d'assurer leur cohérence d'utilisation. Les paramètres de lancement ou les caractéristiques des unités sont obtenus au départ

---

<sup>9</sup> Booch (G.), *Conception orientée objets et applications*.

de fichiers properties afin d'assurer également une grande paramétrisation de l'application.

### III - 3.3. Spécifications réduites :

A partir du menu principal de l'application, un utilisateur peut lancer et visualiser divers serveurs et clients selon les modalités suivantes :

#### *a) Fonction serveur*

Il faut d'abord choisir un numéro de port et sélectionner d'une part un fichier de configuration/description d'une session et d'autre part un fichier d'autorisation. Suite à quoi l'on démarre le serveur qui est alors prêt à accueillir les demandes de connexion des joueurs passées dans les diverses applications clients. Lorsque les joueurs sont connectés, une commande de passage de tour rend actif le premier rôle dont le joueur peut alors passer ses commandes.

#### *b) Fonction client*

Grâce à une interface similaire à celle du serveur, on choisit un serveur hôte en spécifiant son numéro de port ; une fois le serveur contacté, le joueur s'identifie par un nom d'utilisateur et un mot de passe qui sera vérifié d'après le fichier des joueurs chargé par le serveur au démarrage. Le client télécharge alors le contexte de la session active sur le serveur puis démarre. Pour permettre une sélection de la session à laquelle le joueur veut participer, on pourrait avoir recours, dans une implémentation plus complète, à une page html par exemple.

Effets des connexions et déconnexions clients :

disconnect	server en wait
reconnect	idem connect (re-download...)

Le protocole d'échange, asynchrone, est extrêmement simple et de la forme :

{obj}	cmd = string
{obj}	update = string (cmd)

A l'initialisation, tout le contexte est transmis au client, occasionnant une certaine lenteur, surtout dans le cas de sessions fort volumineuses.

#### *c) Modèles de jeu et types de contrôle*

1) Contrôle passif : dans ce modèle, la session, tournant dans son thread,



questionne les différents rôles et attend une commande (il s'agit donc d'une boucle) ; pour un rôle, on a le pseudo-code suivant :

While (his Turn)	
{	
cmd = pCtrl.getCommand();	// pCtrl : Passive Control(read stdin)
game.executeCommand (cmd);	// updates here
}	

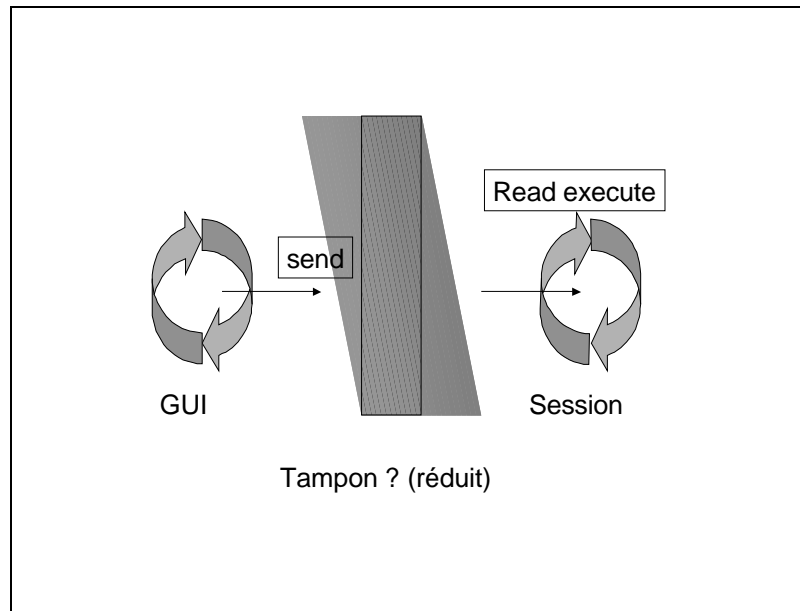
On ne peut donc pas entrer une commande tant que la précédente n'est pas exécutée.

- 2) Interface « réactive » : dans une interface graphique, l'utilisateur produit des événements qui sont traités par un Thread propre ; l'application réagit à ces actions, normalement de façon rapide. En Java, les Events sont traités par des Listeners et c'est là qu'on met en œuvre la logique du programme. La boucle des Events est implicite :

ActionPerformed(evt)	
{	
// it « is » a command :	
cmd = new Command (evt) ;	// first check
game.executeCommand (cmd);	// too long?
}	

Le problème est que la logique est inversée : les actions de la session se déroulent dans le Thread du GUI, pouvant occasionner des délais de réponse trop longs dans le cadre d'une interaction avec l'utilisateur (le principe d'immersion de l'utilisateur se voyant contrarié). A priori, l'utilisateur peut donc aussi activer l'interface à des moments inadéquats, nécessitant par là-même, un contrôle plus strict.

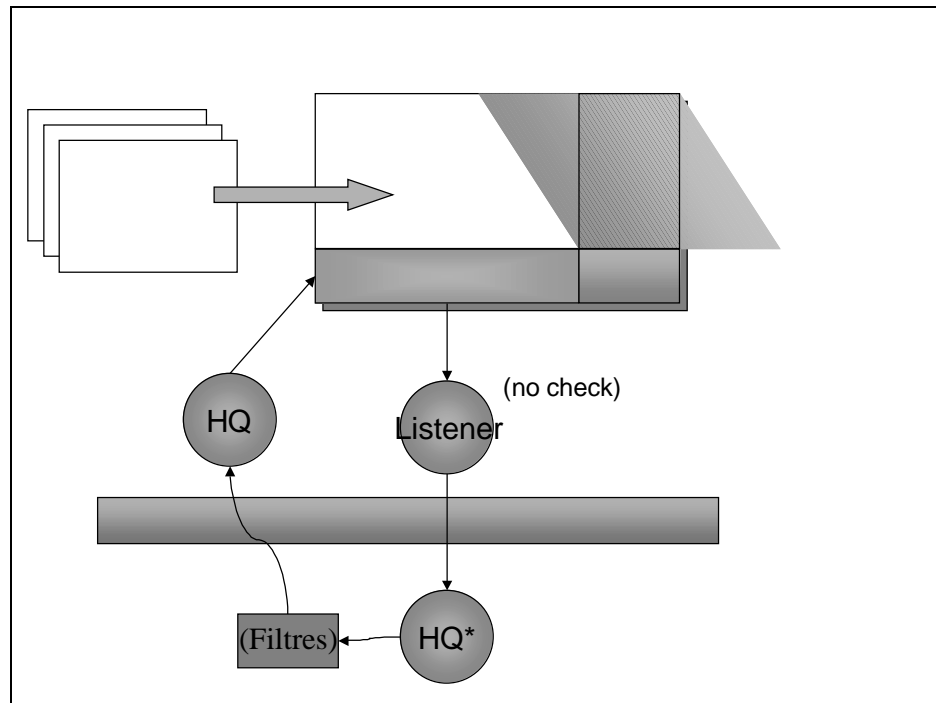
- 3) Comparaison et conciliation des deux approches : si on veut combiner la logique naturelle en boucle de modèle de jeu (contrôleur passif) avec une interface graphique (réactif), on doit imaginer une façon de concilier les deux modes : par exemple, en introduisant un élément intermédiaire assurant la conversion (tampon recevant les actions de l'interface graphique et jouant le rôle de contrôleur passif). Dans le cas du réseau, c'est implicite : quand la commande est délivrée, on a la main à nouveau (asynchrone) ; on doit donc recevoir la notification de la bonne exécution ou de l'échec. Par ailleurs, le mode « instruction » diffère du mode d'« exécution immédiate ».



**Figure 16 : mécanisme de transfert asynchrone**

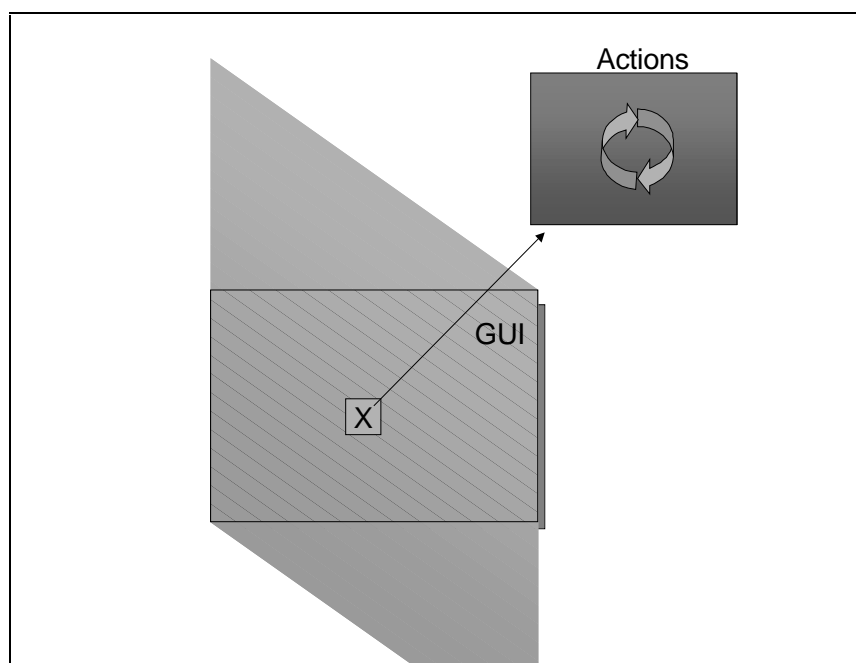
A noter que le caractère synchrone ou non de l'exécution est plus ou moins indépendant du modèle de jeu (tour/tour, temps réel, etc) ; cependant, certaines options sont plus ou moins logiques en fonction du contexte ; en particulier, lié à la perception du joueur. On peut au contraire construire un modèle de jeu « réactif », où des instructions sont envoyées aux unités (avec délai, inconnu, problèmes d'acheminement, etc) et exécutées (avec notification) de façon autonome. Les différents modèles sont alors en fait des règles de gestion de la file d'attente des instructions pour les unités (bloquées en attendant le tour suivant ou exécutées en séquence, etc).

Nous avons retenu l'approche plus simple du tour par tour « passif » avec le lancement d'une commande depuis l'interface graphique (blocage sur tampon de taille 1, implicite sur réseau). Une gestion explicite permettrait d'utiliser un tampon de taille différente (ou infinie).



**Figure 17 : mécanismes de l'interface**

Par ailleurs dans l'implémentation actuelle, nous exécutons la commande dans le thread de l'interface graphique malgré les risques de blocage ou de travail extrêmement long et ce, par souci de facilité. Il nous paraît utile d'expérimenter avec des modèles simples avant de recourir à des gestions plus complexes de l'interaction.



**Figure 18 : interface avec traitement « local »**

### III - 4. Concepts

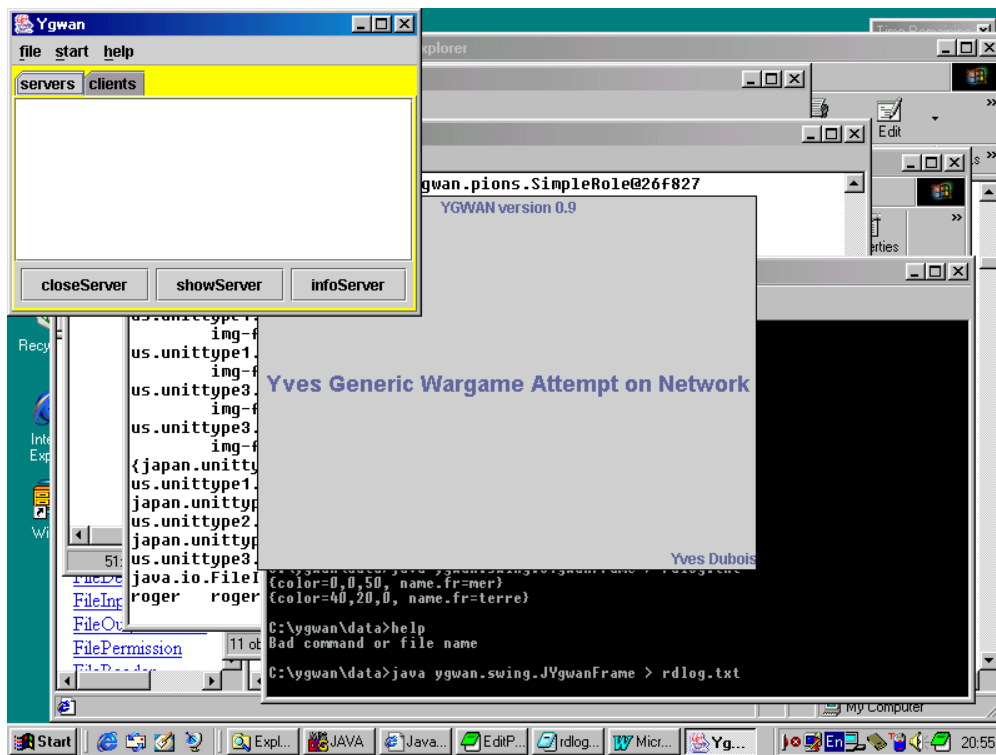


Figure 19 : Vue globale de l'interface initiale

#### III - 4.1. Perception de l'utilisateur

L'interface utilisateur est basée sur les composants standards de swing, plus lourd a priori que awt de base mais disposant de plus de richesse et plus optimisé (double buffering, lightweight components, etc). Nous n'avons pas utilisé de générateur pour les composants de la GUI et ce pour plusieurs raisons : d'abord, nous voulions apprendre à maîtriser cet aspect de la programmation java ; ensuite, nous tenions à conserver un contrôle total de l'implémentation ; enfin, certains assemblages délicats nous semblaient nécessiter un traitement plus fin que celui rendu possible par un générateur.

On aurait bien sûr pu mélanger les deux approches, recourant à un générateur dès que possible, mais dans le cas des classes les plus simples cela ne valait guère la peine tandis que pour les plus complexes l'effort à consentir aurait sans doute été comparable à celui de notre réalisation « manuelle ». S'ajoutant au problème du codage en lui-même, il nous aurait fallu choisir un logiciel et apprendre son utilisation, si bien que les coûts de ce type de développement nous ont semblé plus importants que ses éventuels bénéfices.

Enfin, la « cohabitation » de deux approches aurait pu poser des problèmes (conventions différentes par exemple) rendant le programme plus difficile à déboguer. Par contre, nous avons adopté la philosophie de construction de la plupart des outils de génération de GUI :

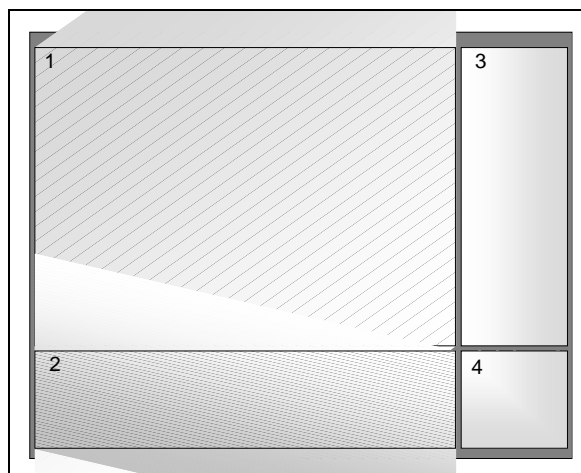
- déclarer les composants dans la classe, afin qu'ils restent accessibles ;

- déléguer la mise en place des composants dans une fonction `initComponents()` appelée par le constructeur.

Ces conventions, par la clarification du code qu'elles permettent, autorisent des modifications plus aisées et systématiques. Pour les composants plus complexes qui implémentent certaines interfaces « réactives » (listeners), nous avons aussi isolé les fonctions relatives aux composants (ex : `resizeComponents()`, etc) et délégué le travail.

Structurellement, la fenêtre de l'interface utilisateur se décompose en quatre panneaux regroupant des fonctionnalités différentes. Le premier d'entre eux (N-O), où se concentre la plus grande partie de l'interaction utilisateur, représente une vue de l'espace de jeu<sup>10</sup>, lui-même décomposé en couches logiques. Le second (S-O) consiste en un simple panneau de log où apparaissent les informations textuelles jugées utiles à l'utilisateur en rapport avec les phases de jeu et les fonctionnalités. Le troisième (N-E) sert à l'affichage des informations relatives aux éléments de l'espace de jeu (informations sur unités, zones, etc). Enfin, le quatrième et dernier panneau regroupe sous forme de boutons les différentes commandes pouvant être passées explicitement par l'utilisateur.

A remarquer : la sélection d'un pion génère le début d'une commande de déplacement de façon automatique (sélection du pion –et donc de la zone de départ-) ; en fonction du degré d'expertise de l'utilisateur, de telles commandes automatiques ou des raccourcis clavier pourraient être multipliés afin de faciliter l'interaction, mais ce point relève de l'ergonomie et nécessiterait donc un examen approfondi pour éviter toute sur-complexification.



**Figure 20 : découpe de l'interface**

---

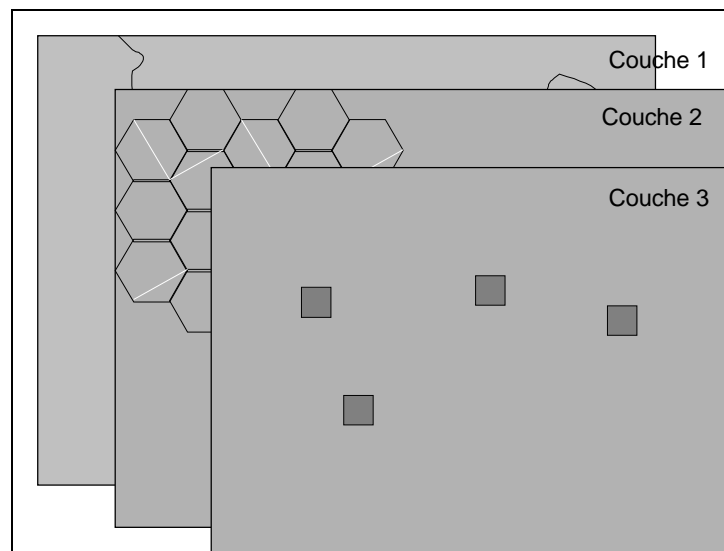
<sup>10</sup> Voir section suivante sur l'espace de jeu.

### a) Représentation de l'espace de jeu

Dans un premier temps, nous avons voulu représenter par des objets graphiques à part entière cellules et zones mais la perte de performance constatée nous a amené à repenser notre approche. Comme, avec les hypothèses faites, seules les zones (éventuellement pleines) ont une signification de jeu, elles seules doivent être modélisées, les cellules ne fournissant qu'une grille de localisation (qu'une représentation graphique pure peut rendre sensible à l'utilisateur) ; cette indication pourrait d'ailleurs être rendue (quoique de façon moins nette) par les zones seules.

En définitive, on peut distinguer trois niveaux (ou couches) superposées dans notre modèle de jeu :

- 1) une représentation purement graphique, visuelle, pour le joueur humain mais qui ne contient pas d'informations utilisées par le jeu ;
- 2) une représentation conventionnelle de l'espace en cellules et zones, homogènes, qui contiennent toute l'information spatiale ;
- 3) une représentation de positionnement des diverses unités (pions) enfin.



**Figure 21 : les différentes couches de l'interface**

Ainsi, nous avons défini une double logique : d'une part, chaque zone -qui représente donc une case atomique- contient la référence au pion qui l'occupe. Par ailleurs, chaque pion référence la zone qu'il occupe, ceci afin d'améliorer l'efficacité du programme.

Pour cette dernière couche, deux cas sont à envisager, selon qu'on accepte plusieurs pions par zone ou pas : dans ce dernier cas, il existe une réelle indépendance entre les pions qui sont seulement localisés ; sinon, on doit plutôt représenter, pour chaque zone, son état en termes d'occupation : vide, un pion, plusieurs pions (de même rôle ou non). Ce choix entraîne donc des logiques de représentation fort éloignées à moins d'utiliser des mécanismes de collation des pions en « piles » présentant des problèmes propres également.

### b) Découpage des hexagones

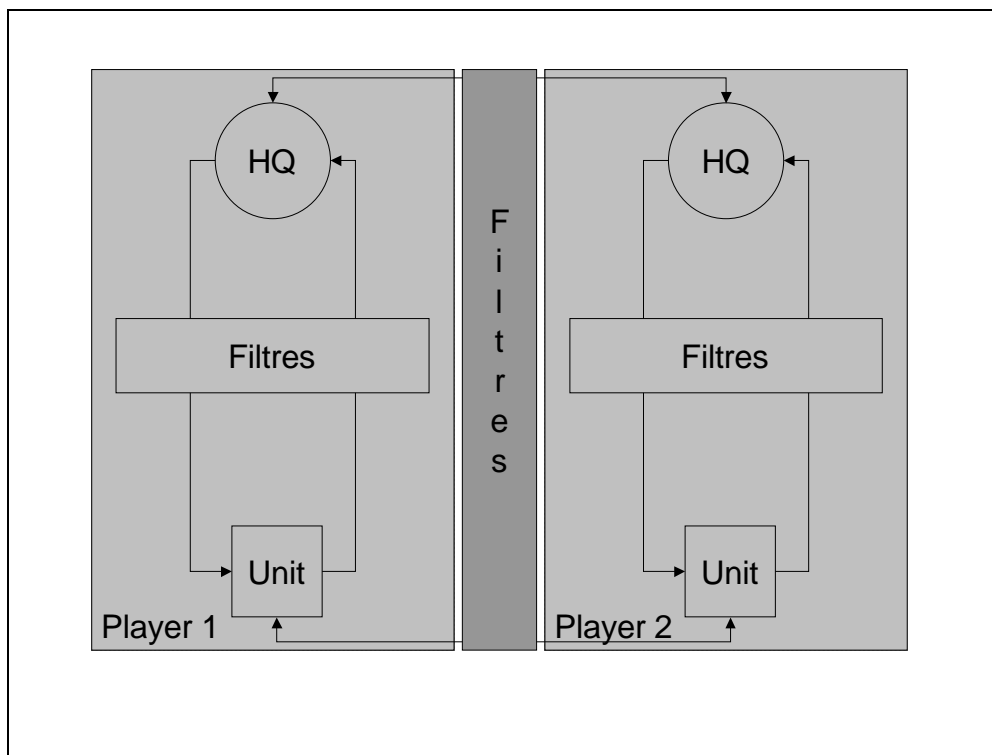
Dans le programme, ces informations sont données dans la classe HexagonGeometry, plus précisément la correspondance entre une lettre et le vecteur d'indices basé en 0 et la correspondance entre le schéma de découpage basé en 0 et la partition correspondante ; le reste est calculé. Par simplicité, nous ferons l'hypothèse qu'un hexagone est toujours complètement divisé en zones simples, c'est-à-dire partitionné. Dans les fichiers de données, on indiquera pour chaque cellule faisant l'objet d'une division son schéma de découpage et pour chacune de ses zones ses propriétés :

**Ex.:** cell.[i,j].split = H0H

zone.[i,j].H0.terrain = sea

zone.[i,j].H3.terrain = land

### c) Filtrage et vue



**Figure 22 : mécanismes de filtre**

En la matière comme en d'autres, nous avons calqué notre représentation sur la réalité, recourant à un mécanisme de filtre, général mais particularisable. La philosophie sous-tendant notre vision des choses est que tout transfert d'information « réel » doit résulter dans un transfert d'information dans notre programme en raison de l'identification entre acteurs réels et objets programmatiques : toute transmission ou communication étant par essence même imparfaite, des possibilités de pertes ou de déformation enrichissent énormément les potentialités du jeu, permettant l'émulation de jeux variés comme la complexification des règles du jeu.

### III - 4.2. Mécanismes de jeu

#### *a) Multi-pions (piles)*

L'utilisation de piles peut donc être un moyen de permettre la présence de plusieurs pions de même rôle dans une même zone, sans changer radicalement de logique par rapport à la présence d'un pion unique grâce à un mécanisme de sur-typage par exemple ; cependant, l'existence d'un tel objet (fortement dynamique en raison des possibles adjonctions ou retraits de pions) oblige à développer des techniques de gestion appropriées.

Quelques actions susceptibles de nécessiter une gestion particulière immédiatement évidentes comprennent l'adjonction d'un nouveau pion, le retrait de l'un d'entre eux de la pile et le choix d'un système de modification des pions constituant la pile, suite à un combat par exemple. Si l'adjonction d'un pion ne présenterait pas nécessairement de problème spécifique puisqu'elle pourrait -par exemple- être résolue comme un sous-cas de l'action « move », le retrait d'un pion de la pile nécessite un mécanisme d'accès et de sélection de chaque pion de la pile. Pour pouvoir ainsi sélectionner les pions de manière individuelle, on pourrait avoir notamment recours à une liste ou encore à un mécanisme de « rotation » du pion actif de la pile.

#### *b) Données et formats*

Dans la réalisation de notre jeu, nous avons dû passer par une phase de génération de données, tant pour définir les propriétés de terrains que pour celles des pions. Dans un souci d'un minimum de réalisme, nous avons choisi de nous baser sur des cartes de la zone Pacifique afin de coller au mieux aux reliefs et accidents de terrain naturels ; par ailleurs, pour les unités, nous avons pris en considération les points forts ou faiblesses que s'accordent à leur reconnaître nombre de stratèges ainsi que la situation stratégique du début du conflit tout en tentant d'établir un certain équilibre afin de préserver la jouabilité. Nous nous sommes également documentés plus copieusement sur les navires et avions, ces deux aspects ayant non seulement été prédominants tout au long du conflit, mais ayant de plus largement expliqué par leurs capacités ou lacunes techniques certains des succès ou des défaites de ce conflit<sup>11</sup>.

D'un point de vue technique, après avoir inséré les données dans un tableur<sup>12</sup>, nous avons sauvegardé ces données sous forme de fichier texte (tab-delimited) avant de les transformer grâce à un programme awk sous forme de fichiers properties. Cette manière de procéder offre à la fois la facilité de manipulation des

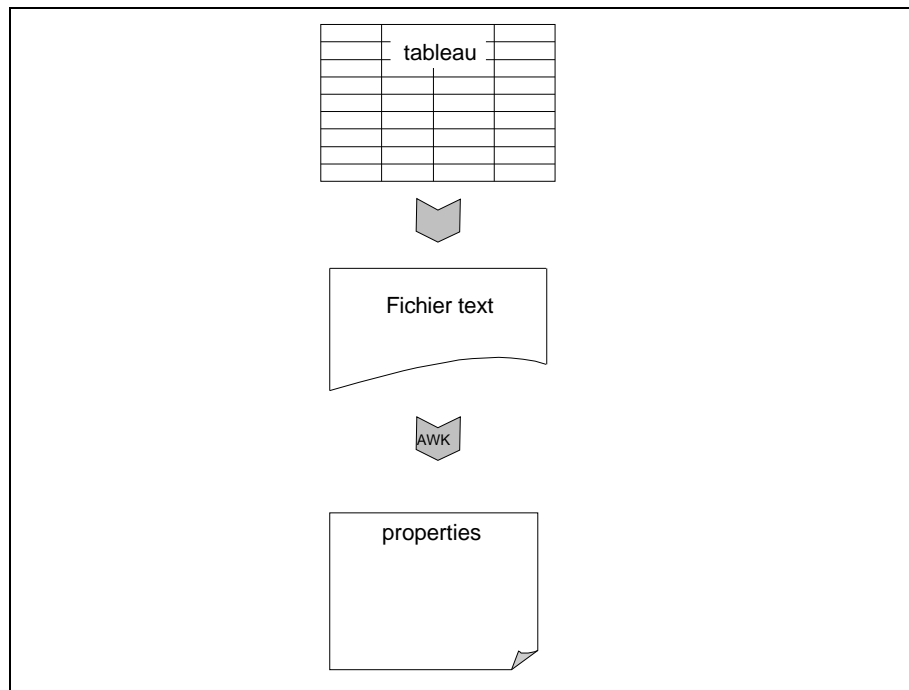
---

<sup>11</sup> Cet aspect de notre travail, peu informatique, nous a cependant demandé un certain temps en raison de la quantité même des données à créer.

<sup>12</sup> Excel en l'occurrence.



fichiers texte par l'utilisateur humain et les avantages du format properties en termes de portabilité et d'homogénéisation des données.



**Figure 23 : système de génération des données**

### *c) Session de jeu*

Une session de jeu est une structure formée de la réunion de plusieurs panneaux : une session est constituée d'un mainHQ et des différents rôles ; le mainHQ contient le GameContext (la carte (image), une grille constituée de cellules (dans notre cas des hexagones), les informations sur le terrain, sur les rôles, etc) et le Field (références pions/zones).

Chaque rôle possède une référence au GameContext et un Field copié/filtré à partir du Field général. Plusieurs objets nécessitent un Thread séparé pour leur exécution. Nous avons choisi en la matière une approche systématique, inspirée de celle de SUN (exemples d'Applets), qui présente au moins deux avantages : d'abord, elle est simple et claire puisqu'on dispose toujours d'une référence sur le Thread mais en ne nous obligeant à implémenter que Runnable ; ensuite, elle permet de gérer le Thread, y compris l'arrêt et le redémarrage, même si nous ne l'avons pas utilisé dans cette implémentation simple.

On équipe l'objet de deux variables internes, une de type Thread, pour conserver le Thread chargé de l'exécution de l'objet (appelé *ownThread* pour cette raison) et une de type boolean pour indiquer si l'exécution doit se poursuivre (souvent, c'est une boucle), appelée *running*.

### III - 4.3. Transmission de l'information

#### *a) Serveur et networking*

Pour chaque demande de connexion, le serveur crée une instance de ClientStub sur le Socket de la nouvelle connexion. Le ClientStub va ensuite être exécuté – évidemment- dans un Thread à part, ce qui permet au serveur de continuer à accepter de nouvelles demandes de connexion (mécanisme serveur réseau classique). Par ailleurs, le serveur garde une trace (Vector) des connexions en cours, ce qui suppose une collaboration –ou du moins un échange d'informations- entre ClientStub et Server.

Chaque ClientStub va se baser sur un SocketTransporter afin de bénéficier de ce qui a été conçu dans ce cadre, en dépit parfois d'une certaine complexité ; rappelons qu'un SocketTransporter lui-même utilise un ObjectWatcher pour surveiller l'arrivée d'objets (asynchrone via Observer/Observable). On est donc amené, avec cette approche (embedded objects), à écrire plusieurs fonctions d'emballage (wrappers). Il serait sans doute plus simple d'implémenter les interfaces directement au niveau de ClientStub mais cela romprait avec la démarche suivie.

Un ClientStub identifie (avec vérification) le joueur autorisé, détermine un rôle dans la session associée au serveur (le premier admissible, ce qui n'est pas ambigu et déterministe) et transmet l'objet HQ correspondant à ce rôle vers le client. Ensuite, il écoute les commandes en provenance du client et envoie vers celui-ci les mises à jour de la situation (sous forme d'autres commandes, en mode texte dans notre exemple réduit, dans cette implémentation).

Amélioration possible : le protocole employé dans notre exemple est très simple et les états de système peu nombreux. La technique employée permettrait un protocole plus riche, avec persistance d'états et journalisation. Par exemple, le déchargement de l'état du jeu (HQ(rôle) sur le serveur) est une opération lourde. On pourrait imaginer de sauver cette information au niveau du client afin de permettre des déconnexions et reconnexions ultérieures en évitant un déchargement complet à nouveau.

Plusieurs variantes sont même possibles : on sauve le HQ initial seulement et on journalise toutes les commandes issues du serveur (sur le client, sur le serveur) ou on sauve l'état du client à certains moments (périodiquement, à la demande, à la déconnexion) et on demande au serveur les commandes passées depuis le dernier point de repère. Nous n'avons pas mis ces mécanismes en œuvre dans notre version simplifiée ; cependant, tous les mécanismes sont en place pour le faire à l'exception de la gestion de la journalisation.

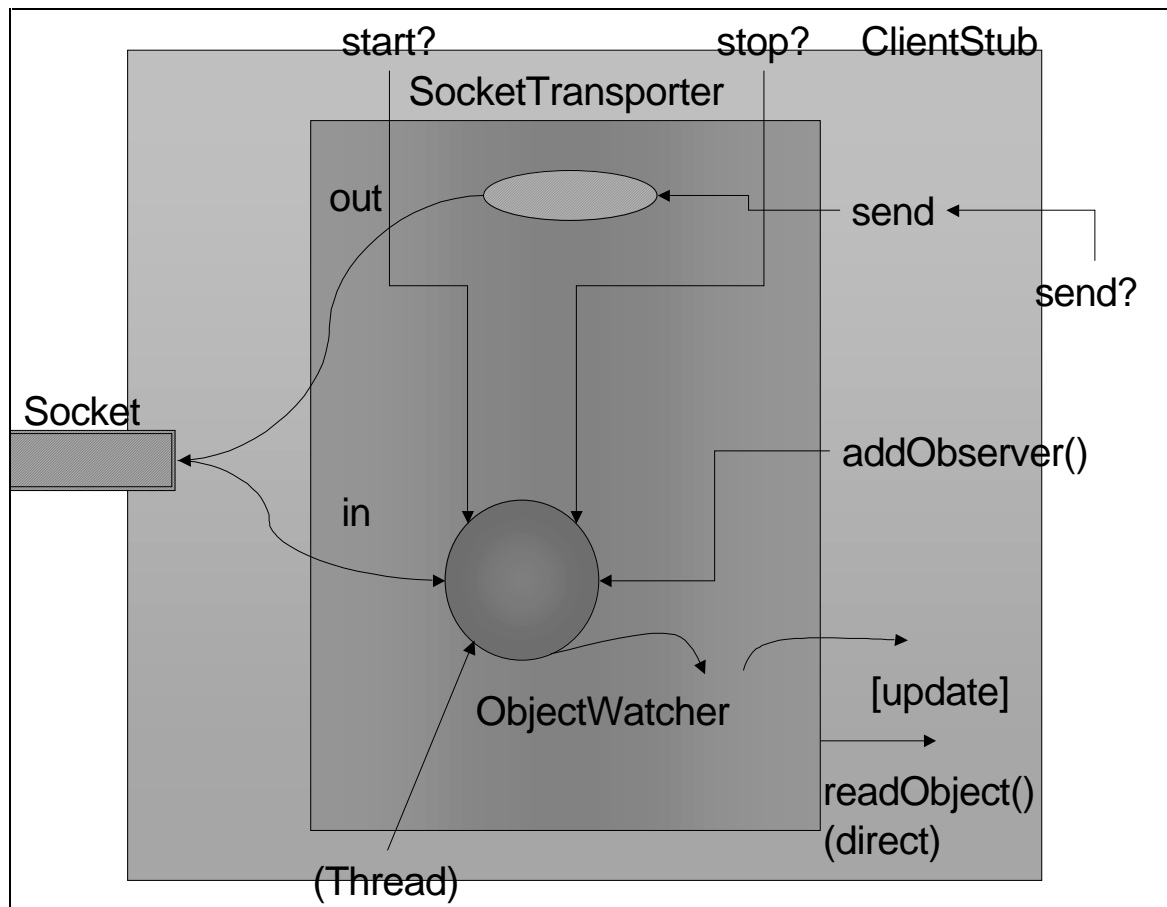


Figure 24 : schéma de fonctionnement de ClientStub

### III - 5. Conclusion

Dans le cadre de cette partie de notre mémoire, nous avons conceptualisé les moyens et structures nécessaires à l'informatisation de notre jeu. Nous avons dû opérer le choix d'un modèle de jeu réduit en raison des contraintes de temps de ce travail ; nous avons choisi de réaliser un jeu de stratégie « classique » avec cellules hexagonales et une interface la plus naturelle possible afin de faciliter l'apprentissage et l'immersion des joueurs.

Pour les mêmes raisons, nous avons choisi le thème de la guerre du Pacifique dans cet exemple jouet. Bien évidemment, cela n'empêche aucunement notre programme d'être utilisé dans d'autres contextes, mais demande simplement la paramétrisation de fichiers propriétés<sup>13</sup>.

<sup>13</sup> Ainsi que le démontre l'utilisation de notre programme par le kot de Mailles dans un contexte médiéval-fantastique.

## **IV. 4<sup>ème</sup> partie : Implémentation<sup>14</sup>**

### ***IV - 1. Introduction***

Dans cette partie, nous étudierons l'aspect purement programmatique de notre travail. Dans un premier temps, nous nous pencherons sur les parties modélisées présentant des difficultés tout en présentant rapidement l'architecture de l'ensemble. Dans une seconde partie, nous tenterons d'ébaucher les fonctionnalités restant à développer et le moyen d'y parvenir selon notre logique de développement. En effet, tous les objectifs ne pourront être atteints dans le cadre limité de ce mémoire et une série de fonctionnalités ne seront réalisées que de manière incomplète.

### ***IV - 2. Concepts***

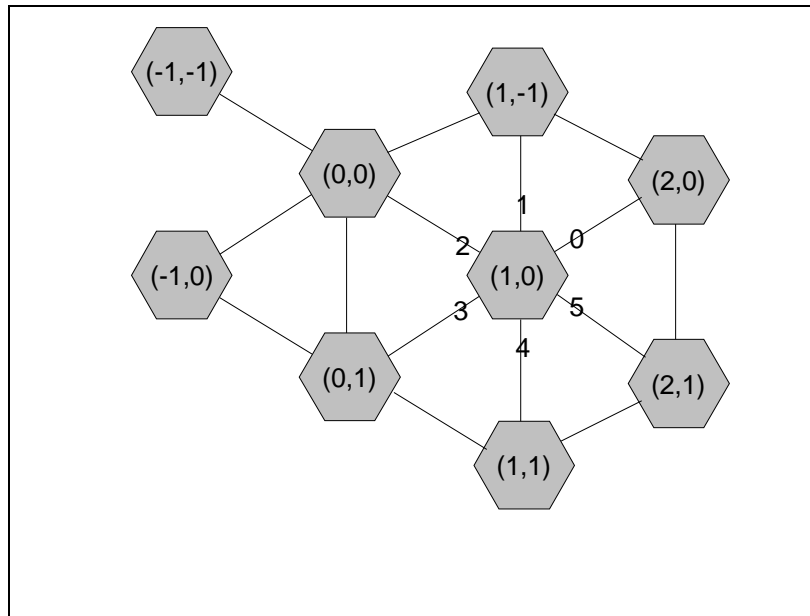
#### **IV - 2.1. Détermination du chemin**

##### *a) Vérification de la contiguïté des hexagones*

Dans Geometry (et donc, en particulier HexagonGEometry), on introduit une fonction donnant la cellule voisine d'une cellule de (ix, iy) donnée, via un segment frontière (de 0 à 5 dans notre cas). Dans notre géométrie hexagonale, on stocke dans la fonction deux tables qui donnent le décalage des cellules voisines en fonction du segment pour les deux cas (ix pair et ix impair) ; ces tables sont construites par inspection de la figure 25.

---

<sup>14</sup> Voir : <http://www.w3.org/> ; également : Subodh Bapat, « *Object-Oriented Networks, models for architecture, operations and management* » ; pour des précisions sur certains termes employés ici, voir : The Free On-Line Dictionary of Computing, <http://wombat.doc.ic.ac.uk/> .



**Figure 25 : Coordonnées relatives des hexagones**

Remarque : la cellule courante n'appartient pas actuellement à la fonction nextCells(cell).

**Tableau 4 : Coordonnées relatives des hexagones**

(0,0) pair		(1,0) impair	
Segment	Delta	Segment	Delta
0	1,-1	0	1,0
1	0,-1	1	0,-1
2	-1,-1	2	-1,0
3	-1,0	3	-1,1
4	0,1	4	0,1
5	1,0	5	1,1

Autres remarques :

- 1) Mécanisme peu performant : boucle sur tous les composants visibles (appelant contains (ix,iy) !). Ainsi, à chaque mouvement de souris, tous les composants sont testés en boucle (swing...)
- 2) Pour une raison inconnue, la fonction setCursor(...) ne donne pas le résultat attendu pour JpolyButton ; une solution palliative est obtenue avec

une Hashtable (si z est dans Ht ET OK(z)      cursor=move sinon  
cursor=normal)

### *b) Etablissement du chemin*

Ce problème se résume au parcours d'un graphe ; dans notre cas, les zones en sont les nœuds. Nous avons décidé de certaines conventions en la matière :

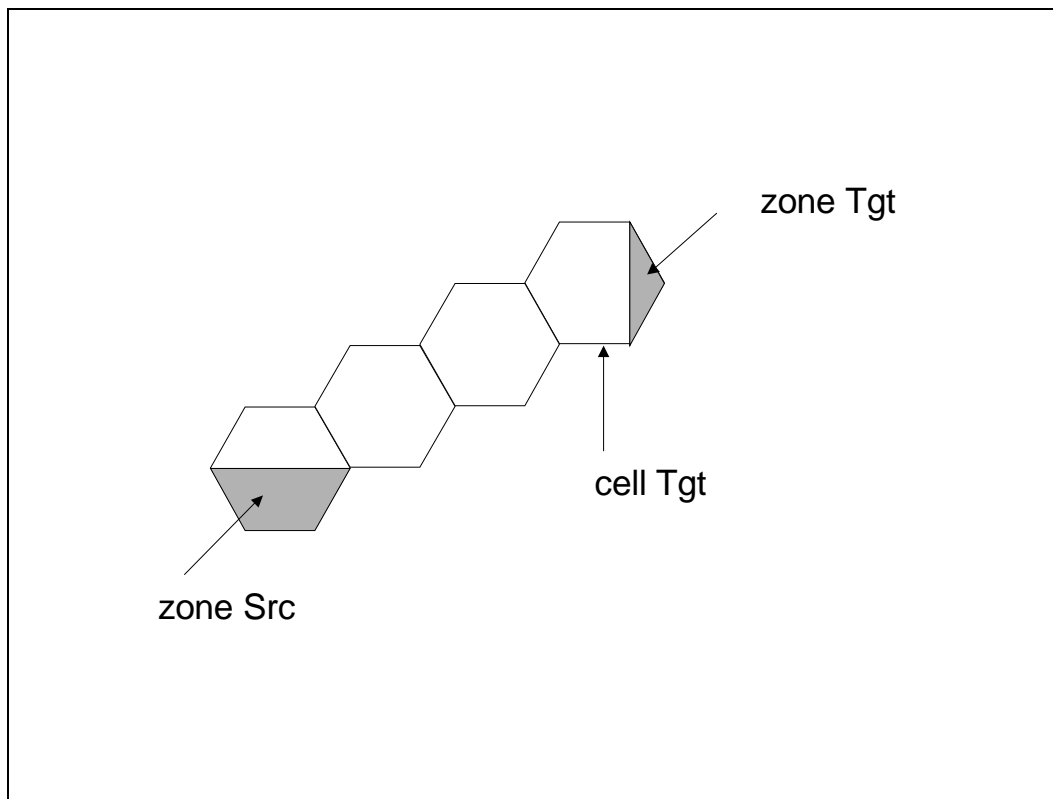
1) la zone source est rendue « clickable » afin que les utilisateurs aient une meilleure perception de l'interaction de sélection-déplacement. En effet, le curseur ne changerait autrement pas de forme (passant de sa forme normale à celle de mouvement) après la sélection du pion. Cependant, la sélection de cette zone comme destination du déplacement entraîne un message de confirmation/infirmation (si confirmation, cela désélectionne le pion).

2) mouvements autorisés : dans un premier temps, seuls les mouvements vers des zones contiguës sont autorisés : il n'y a donc pas de construction de chemin. Plus tard, il serait utile de créer une méthode pour générer le chemin entre zone source et zone de destination (par exemple de la forme générique : `p=Path(src,tgt)` ; `p.build()` , où `src` et `tgt` représentent respectivement la zone source et la zone destination du mouvement)

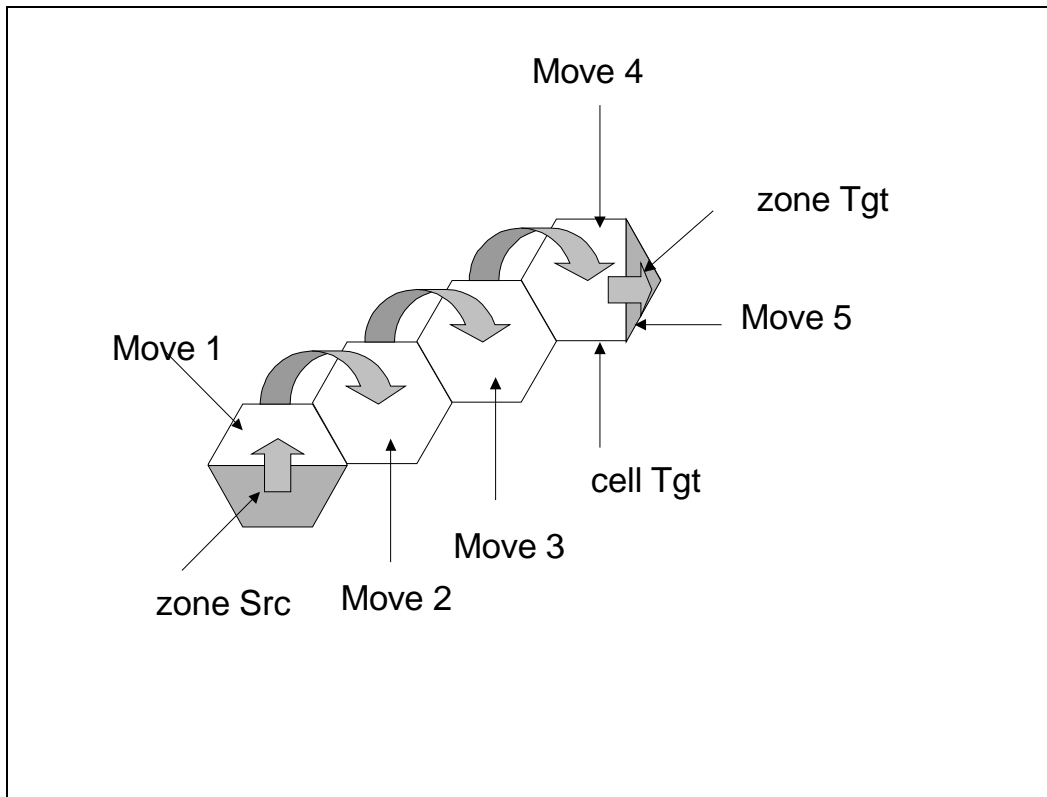
3) vérification de la destination : lorsqu'une commande de mouvement valide est passée, il importe de vérifier la compatibilité du type de terrain de la zone de destination (et des zones intermédiaires dans l'hypothèse d'une construction de chemin) avec les caractéristiques de l'unité. Pour ce faire, nous utilisons une méthode de la forme générale `pion.knowsTerrain(tgt.getTerrain())` dont le résultat positif indique une compatibilité. Dans l'hypothèse d'une construction de chemin, le pseudo-code pourrait être, par exemple :

```
boolean canMove(Pion pion)
{
    Boolean ok = true; // src is supposed OK!
    if (!pion.knowsTerrain(target)) // target ok ?
        return false; // shortcut    speed
    for (int step=0; step<via.size(); step++) // other steps (no shortcut: short path)
    {
        Zone z = (Zone)via.elementAt(step);
        ok &= pion.knowsTerrain(z);
    }
    return ok;
}
```

5) opCost : Le calcul du “opCost” (ou coût en points d’opération) du déplacement devrait être intégré (par exemple à la méthode canMove) mais n’a pas été réalisé dans cette implémentation.



**Figure 26 : établissement d’un chemin avec génération du parcours**



**Figure 27 : établissement d'un chemin case à case**

#### *c) Interaction entre pions*

Dans notre modèle actuel, il ne peut y avoir qu'un seul pion dans une zone donnée ; deux cas seulement vont donc se présenter selon qu'il s'agisse de pions alliés ou ennemis en plus du cas trivial où aucun pion n'occupe la zone de destination :

- 1) il s'agit d'un pion allié (donc d'un pion appartenant au joueur puisque notre exemple ne fait intervenir que deux rôles) : le mouvement n'est pas autorisé (message de mis en garde, le pion reste sélectionné et le joueur peut choisir une autre zone de destination).
- 2) Il s'agit d'un pion ennemi : un dialogue de confirmation de l'attaque est affiché ; si la réponse du joueur est positive, un objet Battle est créé et la bataille résolue.

#### *d) Implications des implémentations du mouvement*

Le choix de l'une ou de l'autre méthode de déplacement entraîne des différences importantes en ce qui concerne l'interaction entre pions :

- 1) Si un pion ne se déplace que d'une zone de départ vers une zone adjacente, lorsqu'un pion ennemi est présent, il est attaqué ; si l'on considère la concaténation de ces déplacements zone à zone contiguë



comme un chemin, tous les pions ennemis dans le chemin sont donc attaqués successivement.

- 2) Si nous autorisons la création de chemins de façon automatisée, de nombreuses logiques sont possibles comme d'ignorer les pions ennemis qui ne seraient pas dans une posture défensive, ou au contraire d'ignorer tous les pions ne possédant pas de facteur de défense contre un type d'unité (cas d'un avion survolant un sous-marin par exemple). De même, il pourrait être autorisé de transiter par une case déjà occupée par un pion allié.

Bien sûr, seule une option qui ignorerait tout pion ennemi présent sur le chemin nous éviterait un travail de vérification supplémentaire, sans quoi il faudra tester la présence de pions ennemis le long du chemin emprunté. Notons que dans ce cas, la commande est soit réduite (option la plus simple) à un déplacement jusqu'à la zone de la première bataille (le pion étant « stoppé » par celle-ci), soit découpé en une série (automatiquement générée) de mouvements partiels ; dans cette dernière éventualité, l'interaction du pion avec les pions ennemis a priori (sans connaître les résultats des batailles intermédiaires) n'est pas claire mais seule la commande concernée est modifiée en tous les cas.

Notons aussi qu'en cas d'un modèle de jeu à information partielle, l'interaction doit être résolue en deux phases, avec des résultats potentiellement différents (au sein du client, durant l'interaction et au sein du serveur sans interaction, le pion est surpris par un pion ennemi non détecté). Avec ceci à l'esprit (et aussi en raison de la simplicité de cette méthode, une commande interactive induisant une seule commande automatique), il semble préférable d'arrêter le mouvement à la première bataille (que ce soit au niveau du client ou du serveur) ; plus clairement : même (voire surtout) avec une implémentation autorisant les mouvements complexes et une information partielle, si un pion se déplace d'une zone à une autre, il est arrêté au premier ennemi sur son chemin (et ce indépendamment de règles optionnelles sur la surprise ou la défense active)

#### *e) Résolution des batailles*

Différentes implémentations sont possibles en la matière et ne diffèrent guère que par les commandes indirectes générées et la complexité des calculs :

0) le pion attaquant gagne toujours, le pion défenseur est retiré du jeu ( méthode très frustrante !)

- 1) un jet de dé (basé dans des versions plus élaborées sur différentes caractéristiques des unités en présence) décide du vainqueur et des conséquences de la bataille (dans le cas d'un système à un seul pion par zone, l'un des deux pions doit donc être repoussé ou détruit). Les résultats possibles en terme de jeu sont :

(p1 attaque p2 dans la zone z2):

- rm p2; mv p1 z2

- rien : p1 est rejeté (pas de mouvement) ; p2 demeure
- rm p1; p1 est détruit, p2 demeure

2) des règles plus complexes permettant des changements de l'état interne des pions (des valeurs de leurs différentes caractéristiques) ; impossible à l'heure actuelle comme nous n'avons pas implémenté une commande « set » (des valeurs des caractéristiques du pion).

## IV - 2.2. Remote Control

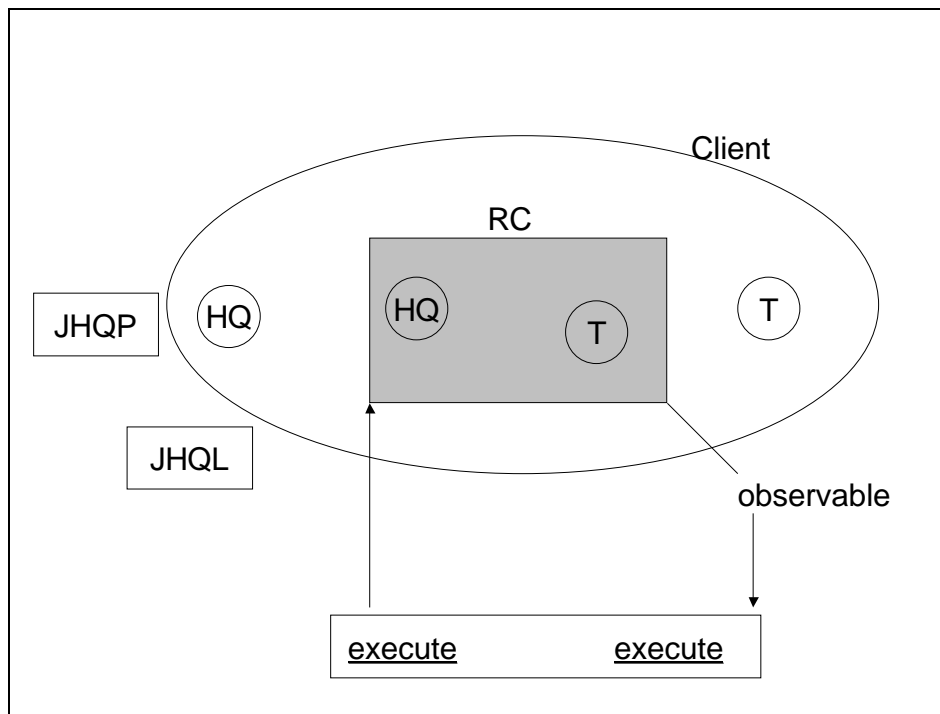


Figure 28 : schéma logique du remote control

## IV - 2.3. Editeur de zones/terrains

### a) Spécifications de l'éditeur de zones/terrains

Pour qu'un éditeur mérite ce nom, il doit être capable de sauvegarder/charger des fichiers et de modifier les informations lorsque celles-ci sont chargées en mémoire. Dans la version actuelle, l'information est répartie dans différents fichiers, à savoir `cells.properties` (pour les cellules, comprenant les informations de découpe) et `zones.properties` (pour les différentes zones définies par la découpe). De plus, une zone hérite des propriétés de sa cellule mère, induisant la possibilité d'un conflit à ce niveau.

Plus précisément, quelles sont les données conservées et celles qui sont perdues lors de l'altération du modèle de découpe d'une cellule ? Bien sûr, les propriétés de la cellule demeurent inchangées mais toutes les propriétés des zones de la cellule

sont ramenées à leur valeur par défaut. Les propriétés des types de terrain sont bien évidemment utilisées mais ne sont pas éditées dans cette version.

Lorsque l'édition est terminée, les propriétés des zones et des cellules sont sauvegardées dans différents fichiers ou dans un fichier unique (en fonction de l'option portant sur le préfixe des fichiers properties) ; elles peuvent être regroupées par cellules, par zone ou par nom de propriété, permettant plus aisément différents types de traitements ultérieurs. Les propriétés des zones de cellules non modifiées sont conservées inchangées.

### ***IV - 3. Structure et découpage en packages***

Nous allons reprendre dans cette partie les différentes classes de notre programme subdivisées selon leurs packages, puis nous reprendrons le code d'une classe à titre exemplatif.

La découpe en packages que nous avons définie présente certaines limitations<sup>15</sup>, notamment l'existence de nombreux liens fonctionnels entre packages ; ce défaut provient principalement de l'adoption de compromis dans la manière de structurer les différentes fonctionnalités, certaines convenant mieux à une découpe purement fonctionnelle, d'autres à une approche résolument objet. Nous avons principalement favorisé la clarté du code, au détriment parfois d'une certaine homogénéité.

Dans la présentation des classes, nous reprenons la ligne de déclaration et celles des classes "embedded". Cette façon de faire, bien que loin d'être sans défaut, permet tout à la fois une lecture aisée, la mise en évidence des héritages et interfaces et des regroupements logiques des classes et concepts. Nous renvoyons au code des sources (annexe séparée) pour plus de détails ; nous avons en effet très diversement documenté les classes en fonction de leur complexité ou de leur importance.

#### **IV - 3.1. Package ygwan**

En principe, charge "client" et "serveur" ; devrait faire appel à JygwanFrame (appel direct aujourd'hui). Ce package regroupe aussi les classes d'usage général.

##### **Field.java**

```
public class Field implements Serializable, GameControl
```

Particularités et fonctionnalités :

§ Where

§ What

---

<sup>15</sup> sans compter la taille très inégale des classes et packages.

```

§ Pions = {}

§ Cmd      rm

          mv

```

### **GameContext.java**

```
public class GameContext implements Serializable
```

Les informations sous-jacentes à la partie, un regroupement logique.

### **GameControl.java**

```
public interface GameControl
```

Interface servant à l'exécution des commandes.

Ex : GameSession, HQ, Field, RemoteControl,...

Il y a délégation d'exécution de ClientStub    Session    HQ    Field.

Il serait intéressant de donner une variante avec Command, classe plus structurée qu'un simple String (une analyse serait à prévoir)    execute (Command cmd).

```
Execute (String cmd) = { Command cmd = Command.createCommand (cmd) ;
execute (cmd) }
```

### **GameModel.java**

```
public interface GameModel
```

Idée initiale GameModel.run()

- boucle sur les tours
- boucle sur les rôles    contrôleur passif

On a changé d'optique : cette classe n'est pas utilisée, mais reste au cas où on change vers un contrôleur passif (cf adaptation)

### **GameSession.java**

```
public class GameSession implements Serializable, GameControl
```

Partage en différents niveaux    Session State ?

roles Seq    tout à tour (dans une future implémentation en random ?)

findRole (player)    may duplicate

save/load : nécessaire de sauvegarder et récupérer (par exemple en démarrant seulement à partir d'une session sauvée, préparée par « parse ») ; pas implémenté.

## HQ.java

```
public class HQ extends Observable implements Observer, Serializable,
GameControl
```

Il s'agit d'une classe très importante : info server, propagation vers client

Quelques fonctionnalités :

§ Execute(GameControl)

§ Update(Observer)

§ Observable

§ Gestion des rôles (turns)

## Player.java

```
public class Player
```

Cette classe s'occupe de l'authentification des joueurs, identifiés par leur nom d'utilisateur (userid) et leur mot de passe (pwd), avec présence d'un troisième champ optionnel avec le nom du joueur. Actuellement, ces renseignements sont stockés dans un fichier text (tab delimited) en clair. Il va sans dire qu'il conviendrait de sécuriser un minimum cette partie de notre programme (même un hashCode() conjointement à equals()) serait déjà une amélioration). Dans le même ordre d'idées, si un utilisateur est renseigné dans le fichier sans mot de passe, seule l'absence de mot de passe (mot de passe vide) permettra l'autorisation de l'utilisateur.

## ResourceClass.java

```
public class ResourceClass
```

Sert à la recherche de ressources relatives aux classes (utilité dans le cadre d'un jar par exemple). Sa seule fonctionnalité est d'exister, ce qui explique son code des plus légers ; ce rôle pourrait d'ailleurs être joué par Ygwan lui-même.

## Ygwan.java

```
public class Ygwan extends JFrame
    public class AL implements ActionListener
```

## IV - 3.2. Package ygwan.grid

### Cell.java

```
public interface Cell extends java.io.Serializable
```

### Geometry.java

```
public abstract class Geometry implements java.io.Serializable
```

Encapsule la géométrie et sert à favoriser le code indépendant; à noter une astuce pour améliorer les performances: un cache, couplé à un mécanisme de "lazy evaluation".

#### **GeoUtil.java**

```
public class GeoUtil implements java.io.Serializable
```

#### **Grid.java**

```
public class Grid implements java.io.Serializable
```

#### **GridParser.java**

```
public class GridParser
```

#### **HexagonGeometry.java**

```
public class HexagonGeometry extends Geometry
```

#### **Path.java**

```
public class Path
```

#### **SimpleCell.java**

```
public class SimpleCell implements Cell
```

#### **SimpleZone.java**

```
public class SimpleZone implements Zone
```

#### **Zone.java**

```
public interface Zone extends java.io.Serializable
```

### **IV - 3.3. Package ygwan.network**

#### **Client.java**

```
public class Client
```

#### **ClientStub.java**

```
public class ClientStub implements Observer, Runnable // extends Thread ?
```

#### **ControlObserver.java**

```
public class ControlObserver implements Observer
```

Exemple d'adaptateur (Design Patterns).

#### **ObjectWatcher.java**

```
public class ObjectWatcher extends Observable implements Runnable
```

(readObject()) explicite, synchronize

### RemoteControl.java

```
public class RemoteControl extends Observable implements Observer,  
GameControl
```

### SenderObserver.java

```
public class SenderObserver implements Observer
```

Exemple d'adaptateur (Design Patterns).

### Server.java

```
public class Server implements Runnable
```

### SocketTransporter.java

```
public class SocketTransporter extends Transporter implements Observer
```

Transporte les données sur un socket (un des mécanismes possibles; le seul implémenté). A noter l'usage de reset(), nécessaire pour éviter une grande utilisation de la mémoire (car java note les objets écrits dans un ObjectOutputStream).

### Transporter.java

```
abstract public class Transporter extends Observable
```

Classe abstraite et non interface à cause de la classe Observable.

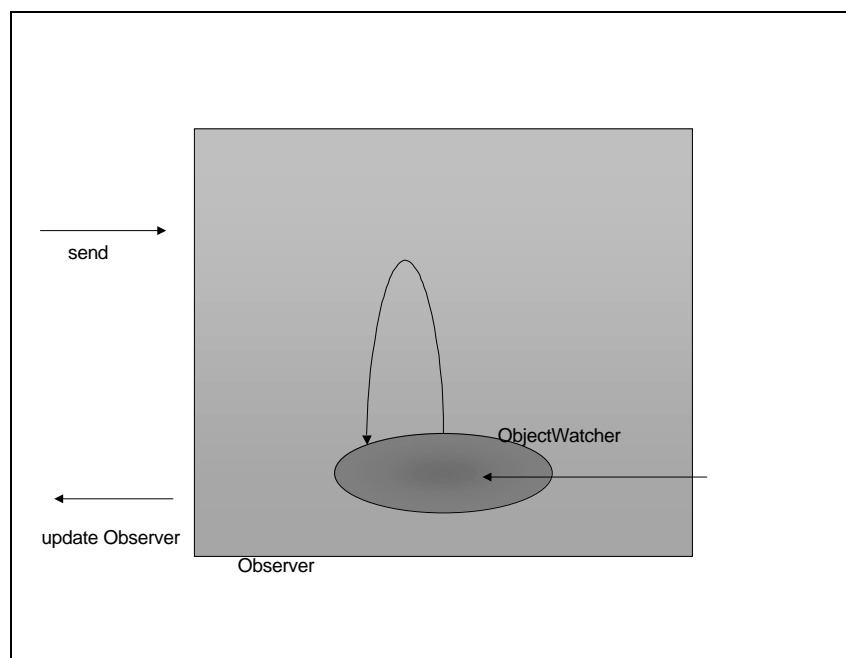


Figure 29 : fonctionnement de l'observer

### **TransporterClient.java**

```
public class TransporterClient extends Transporter
```

Pas utilisé (correspond à la toute première implémentation, pas symétrique).

### **TransporterEchoServer.java**

```
public class TransporterEchoServer
```

### **TransporterServer.java**

```
public class TransporterServer extends Transporter
```

## **IV - 3.4. Package ygwan.pions**

### **Battle.java**

```
public class Battle implements Runnable
```

### **Command.java**

```
public class Command
```

Pas encore utilisé, permettrait d'éviter les analyses de string répétées (erreurs, maintenance, etc)

### **Pion.java**

```
public interface Pion extends java.io.Serializable, Cloneable
```

### **PionFactory.java**

```
public interface PionFactory extends java.io.Serializable
```

### **Role.java**

```
public interface Role extends java.io.Serializable
```

### **SimpleBattle.java**

```
public class SimpleBattle extends Battle
```

### **SimplePion.java**

```
public class SimplePion implements Pion
```

Une double structure de donnée existe pour SimplePion : sous forme de Properties (provenant des fichiers) et comme variables dans l'objet (cohérence).

### **SimplePionFactory.java**

```
public class SimplePionFactory implements PionFactory
```

### **SimpleRole.java**



```
public class SimpleRole implements Role
```

### SimpleRoleParser.java

```
public class SimpleRoleParser
```

## IV - 3.5. Package ygwan.swing

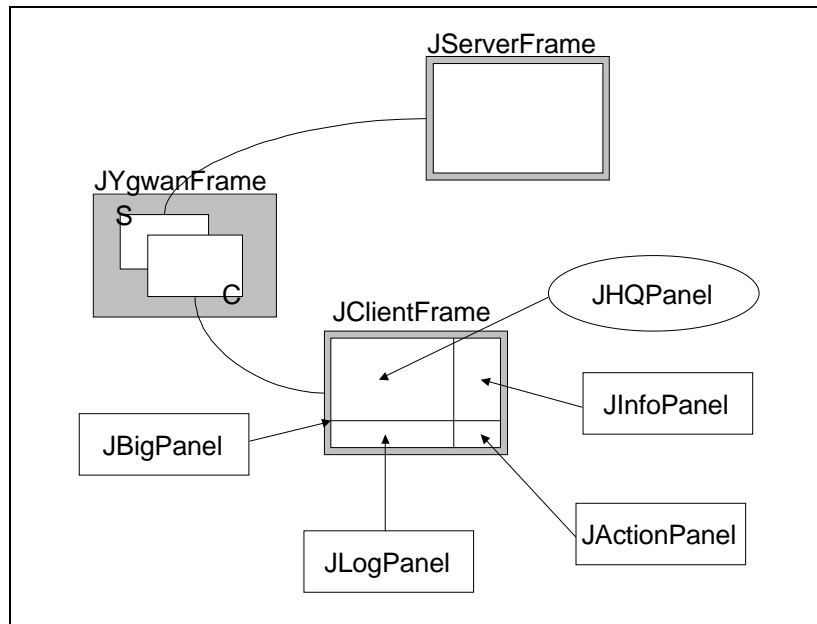
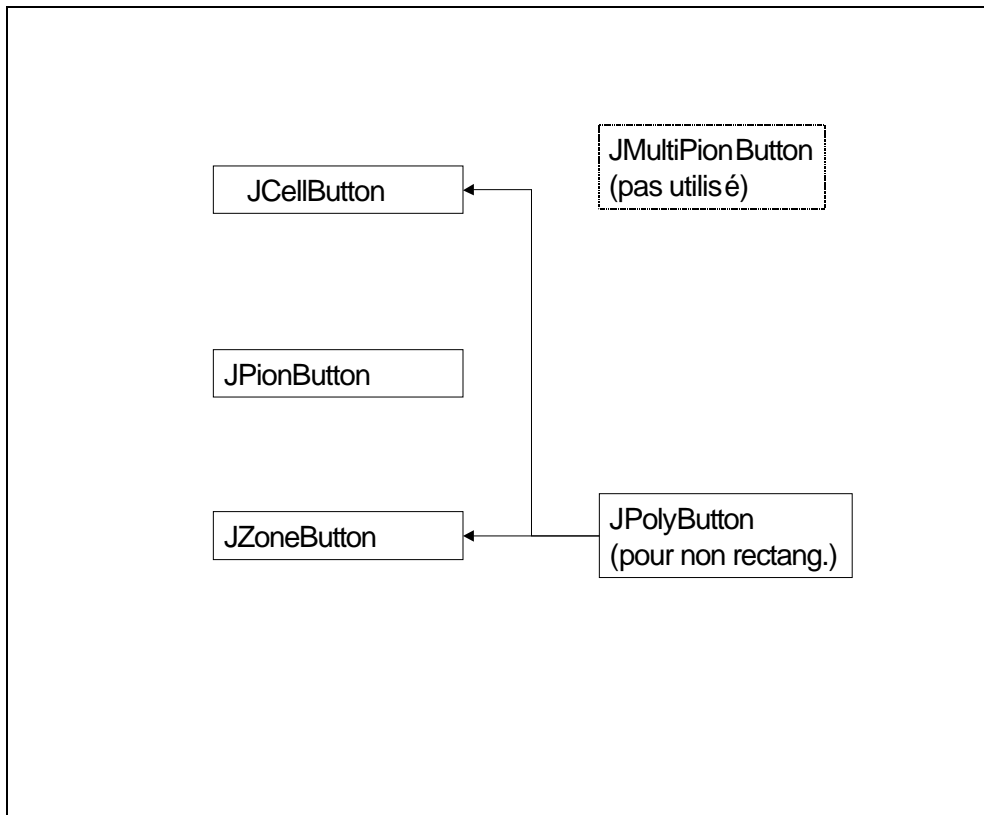


Figure 30 : découpage en classes de l'interface



Figure 31 : Vue de l'interface générale



**Figure 32 : structure des classes des boutons**

### **JActionPanel.java**

```
public class JActionPanel extends JPanel
```

### **JBigPanel.java**

```
public class JBigPanel extends JSplitPane
```

### **JCellButton.java**

```
public class JCellButton extends JZoneButton
```

### **JClientFrame.java**

```
public class JClientFrame extends JFrame
```

### **JEditActionPanel.java**

```
public class JEditActionPanel extends JPanel
```

### **JEditorFrame.java**

```
public class JEditorFrame extends JFrame
    public class AL extends WindowAdapter
    public class ML implements ActionListener
```

### **JEditPanel.java**

```
public class JEditPanel extends JHQPannel implements ActionListener
```

### **JFieldView.java**

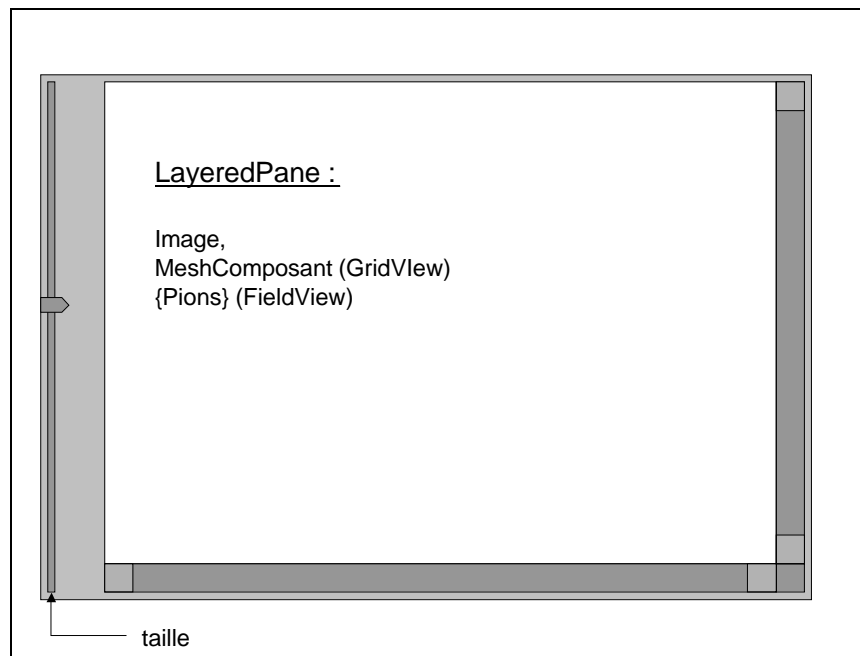
```
public class JFieldView
```

### **JGridView.java**

```
public class JGridView
```

### **JHQPanel.java**

```
public class JHQPanel extends JPanel implements ChangeListener, Observer
```



**Figure 33 : Construction du JHQPanel**

### **JImgLabel.java**

```
public class JImgLabel extends JLabel implements ChangeListener
```

### **JInfoPanel.java**

```
public class JInfoPanel extends JPanel implements PropertyChangeListener
```

### **JLogPanel.java**

```
public class JLogPanel extends JPanel implements Logger  
    public class AL implements ActionListener
```

### **JMeshComponent.java**

```
public class JMeshComponent extends JComponent
```

### **JMiniBrowserPanel.java**

```
public class JMiniBrowserPanel extends JPanel implements HyperlinkListener,  
ActionListener
```

### **JMultiPionButton.java**

```
public class JMultiPionButton extends JButton
```

### **JPionButton.java**

```
public class JPionButton extends JButton
```

### **JPolyButton.java**

```
public class JPolyButton extends JButton
```

### **JServerFrame.java**

```
public class JServerFrame extends JFrame
```

Seules des différences mineures distinguent JServerFrame et JclientFrame.

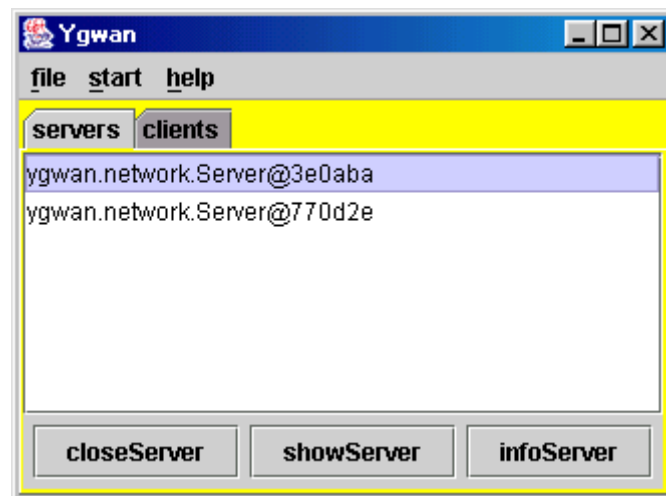


Figure 34 : JServerFrame

### **JYgwanFrame.java**

```
public class JYgwanFrame extends JFrame
    public class AL extends WindowAdapter implements ActionListener
```

### **JZoneButton.java**

```
public class JZoneButton extends JPolyButton
```

## **IV - 3.6. Package ygwan.swing.dialogs**

### **JAboutDialog.java**

```
public class JAboutDialog extends JDialog
```

### **JAboutPanel.java**

```
public class JAboutPanel extends JPanel
    public class HLL implements HyperlinkListener
```

## JClientDialog.java

```
public class JClientDialog extends JDialog implements ActionListener
    public class CL implements javax.swing.event.ChangeListener
```

## JServerDialog.java

```
public class JServerDialog extends JDialog implements ActionListener
    public class FC implements ActionListener
    public class CL implements javax.swing.event.ChangeListener
```

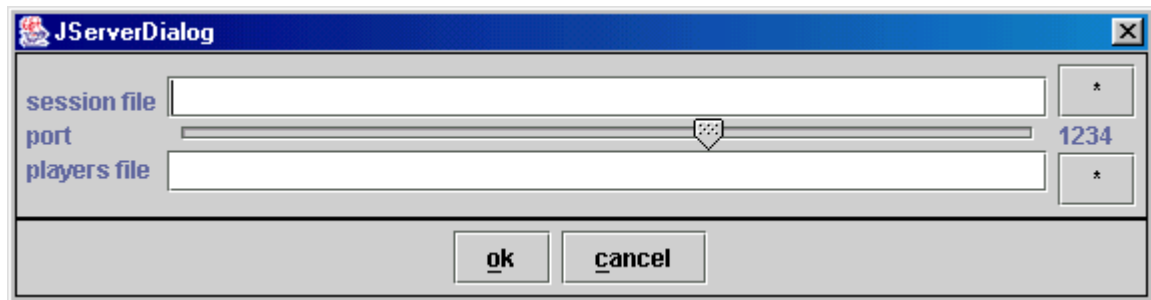


Figure 35 : JServerDialog

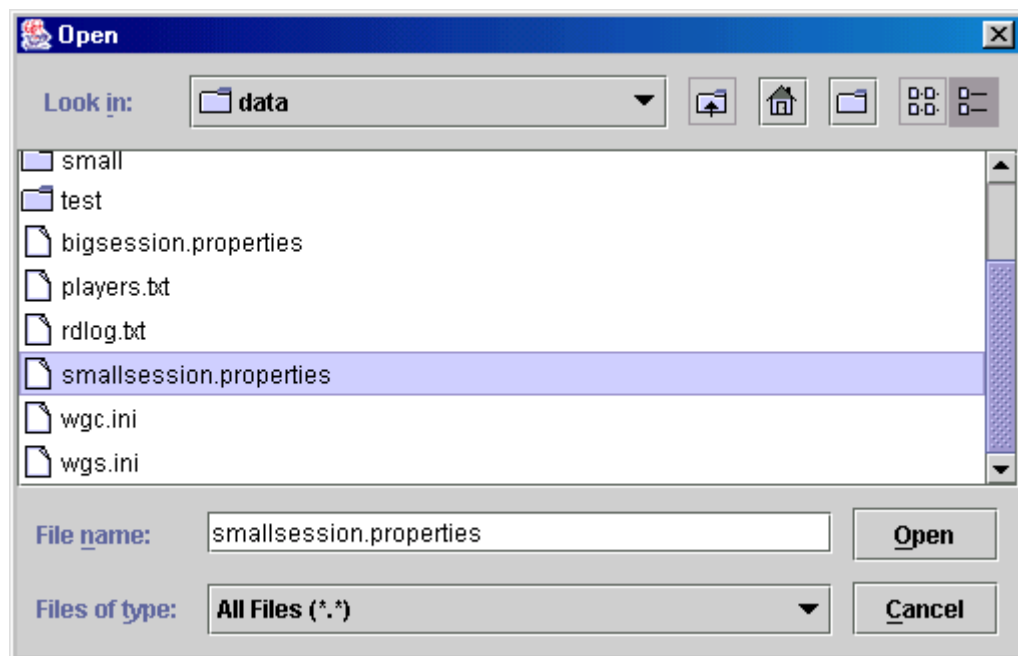


Figure 36 : Menu de sélection de fichiers

## JSplashPanel.java

```
public class JSplashPanel extends JPanel
```

## JSplashWindow.java

```
public class JSplashWindow extends JWindow
```

## JSplitZoneDialog.java

```
public class JSplitZoneDialog extends JDialog implements ActionListener
```

### **JTerrainDialog.java**

```
public class JTerrainDialog extends JDialog implements ActionListener
```

## **IV - 3.7. Package ygwan.swing.listeners**

### **JHQListener.java**

```
public class JHQListener extends MouseAdapter implements ActionListener
```

### **LogListener.java**

```
public class LogListener implements Observer, ActionListener
```

### **YHyperLinkListener.java**

```
public class YHyperLinkListener implements HyperlinkListener
```

## **IV - 3.8. Package ygwan.test**

### **DiceTest.java**

```
public class DiceTest
```

### **GameSessionTest.java**

```
public class GameSessionTest
```

### **GameWorldTest.java**

```
public class GameWorldTest
```

### **GeometryTest.java**

```
public class GeometryTest
```

### **GridTest.java**

```
public class GridTest
```

### **JAboutDialogTest.java**

```
public class JAboutDialogTest
```

### **JAboutPanelTest.java**

```
public class JAboutPanelTest
```

### **JBigPanelTest.java**

```
public class JBigPanelTest
```

### **JHQPanelTest.java**

```
public class JHQPannelTest
```

### **JImgLabelTest.java**

```
public class JImgLabelTest
```

### **JLogPanelTest.java**

```
public class JLogPanelTest
```

### **JPolyButtonTest.java**

```
public class JPolyButtonTest
```

### **JSplashPanelTest.java**

```
public class JSplashPanelTest
```

### **JSplashWindowTest.java**

```
public class JSplashWindowTest
```

### **JustATest.java**

```
public class JustATest
```

### **PlayerTest.java**

```
public class PlayerTest
```

### **PrefixPropertiesFilterTest.java**

```
public class PrefixPropertiesFilterTest
```

### **PropertiesLoaderTest.java**

```
public class PropertiesLoaderTest
```

### **ResourceClassTest.java**

```
public class ResourceClassTest
```

### **SimplePionFactoryTest.java**

```
public class SimplePionFactoryTest
```

### **SimpleRoleTest.java**

```
public class SimpleRoleTest
```

### **TerrainTest.java**

```
public class TerrainTest
```

## IV - 3.9. Package ygwau.util

### **Dice.java**

```
public class Dice
```

### **EndingStringFilter.java**

```
public class EndingStringFilter implements Filter
```

### **Filter.java**

```
public interface Filter
```

PrefixStringFilter, EndingStringFilter, FilteredEnumeration, PrefixPropertiesFilter.

### **FilteredEnumeration.java**

```
public class FilteredEnumeration implements Enumeration
```

### **Loader.java**

```
public class Loader
```

Pas utilisé.

### **LogGenTimer.java**

```
public class LogGenTimer extends javax.swing.Timer  
    public class TAL implements ActionListener
```

### **Logger.java**

```
public interface Logger
```

LogGenTimer, PrintStreamLogger et (appartenant au package swing) JlogPanel.

### **MultiHashtable.java**

```
public class MultiHashtable extends Hashtable  
    class MultiEnum implements Enumeration
```

### **PrefixPropertiesFilter.java**

```
public class PrefixPropertiesFilter
```

### **PrefixStringFilter.java**

```
public class PrefixStringFilter implements Filter
```

### **PrintStreamLogger.java**

```
public class PrintStreamLogger implements Logger
```

### **PropertiesLoader.java**



```
public class PropertiesLoader
    public class SST
```

Permet la lecture à partir de différents formats de fichiers (fonctionnalité pas utilisée).

La classe auxiliaire SST n'est qu'un String Tokenizer « simplifié » ; en effet, « a \t \t b » est interprété par le StringTokenizer comme « a b » alors que « a \t \t b » interprété par SST fournit a null b, comme nécessaire ici .

### **StopWatch.java**

```
public class Stopwatch
```

## **IV - 3.10. Package ygwan.world**

### **GameWorld.java**

```
public class GameWorld implements java.io.Serializable
```

### **GameWorldParser.java**

```
public class GameWorldParser
```

### **Terrain.java**

```
public class Terrain extends Properties implements java.io.Serializable
```

### **TerrainParser.java**

```
public class TerrainParser
```

Les divers parsers pourraient être regroupés dans ygwan.util, en factorisant certaines actions dont le code est répété.

## **IV - 3.11. Exemple de code**

A titre d'exemple, voici le code de la classe ObjectWatcher (package ygwan.network), intéressante en particulier pour la gestion du parallélisme (Thread) :

```
package ygwan.network;
```

```
import java.io.*;
```

```
import java.util.*;
```

```

{
    ObjectInputStream objInputStream = null;
    Thread ownThread = null;
    boolean running = false;

    public ObjectWatcher(ObjectInputStream objectInputStream)
    {
        this.objInputStream=objInputStream;
        System.err.println(this+" ready; ois="+objInputStream);
    }

    public synchronized Object readObject() throws Exception
    {
        Object obj = null;
        System.err.println("reading in "+this);
        obj = objInputStream.readObject();
        System.err.println(this+" : obj="+obj);
        return obj;
    }

    public void start()
    {
        if (ownThread==null)
            {ownThread = new Thread(this);}
        ownThread.start();
    }

    public void stop()

```

```

public void run()
{
    running = true; // just in case

    try
    {
        while(running)
        {
            Object received = this.readObject();
            setChanged();
            notifyObservers(received);
            // ois.reset(); // ???
        }
    }

    catch (Exception ex)
    {
        System.err.println(ex);
        notifyObservers(ex); // ???
    }

    ownThread = null; // more cleaning? also finalize() ?
}
}

```

#### ***IV - 4. Conclusion***

Notre programme informatique est arrivé au terme de cette première phase de réalisation ; nous avons réalisé un exemple de jeu fonctionnel aux spécifications réduites. Dans ce cadre, nous avons préféré le développement de classes diversifiées à une implémentation plus poussée d'un nombre limité de classes afin

d'obtenir sinon un programme parfaitement jouable, au moins un exemple fonctionnel. Par ailleurs, soit par souci d'esthétisme informatique, soit à cause des changements apportés à notre vision en cours de développement, nous avons parfois développé certaines classes peu ou pas utiles directement à notre propos.

# **Conclusion générale**

## ***Objectifs atteints***

Ce travail nous a permis de démontrer la faisabilité de la programmation d'un jeu de stratégie réseau en java ; les axes principaux de ce développement ont été :

La portabilité de l'application, permise notamment par le développement en java (machine virtuelle java) et par l'utilisation de formats standards pour nos fichiers.

La généricité de l'application obtenue par le choix d'un modèle de jeu ouvert, permettant la modélisation de situations variées.

La facilité de développement et d'amélioration de notre programme, grâce à une architecture modulaire et au recours à des fichiers de paramétrisation.

La réalisation d'une version démo opérationnelle, illustrant les différentes fonctionnalités étudiées dans ce travail.

## ***Difficultés et apprentissage***

Ce travail nous a permis de parfaire notre connaissance de java, et particulièrement des classes relatives à l'interface graphique (swing) ; ce processus de développement / apprentissage a lui-même entraîné son lot de difficultés puisque nous avons parfois découvert de nouveaux concepts plus efficaces à des phases ultérieures du développement, expliquant une certaine hétérogénéité dans les solutions apportées dans différentes classes à des problèmes de nature similaire.

Cela nous a aussi poussés à utiliser des classes parfois trop complexes par rapport aux besoins de notre exemple dans un double souci d'extensibilité des fonctionnalités et d'apprentissage : ainsi pour l'utilisation de TabPanel pas très utile dans notre interface.

Des difficultés imprévues ont également surgi, particulièrement en matière d'interface : ainsi, les performances médiocres des méthodes natives de swing lorsque le nombre de boutons présents est important (parcours des boutons).

## ***Améliorations futures***

Il conviendrait d'apporter un certain nombre d'améliorations à notre programme, en développant de nouvelles classes mais aussi en étendant les fonctionnalités de plusieurs classes que nous avons développées.

Ainsi, les performances comme l'efficacité du programme pourraient être accrues par le recours à un serveur http pour applications web.

De même, il faudrait permettre la création d'unités en cours de jeu et la modification des propriétés des hexagones.

On pourrait également implémenter un algorithme de parcours de graphe pour permettre des déplacements vers des cellules non-limitrophes.

Par ailleurs, la scénarisation et les mécanismes de gestion et de paramétrisation du jeu devraient être développés. Plusieurs idées permettraient d'améliorer celui-ci : plus grand nombre d'unités présentes, renforts générés selon tel ou tel facteur, meilleures formules d'attaque (plus réalistes ou équilibrées), etc mais la plupart de ces améliorations demandent de compléter les mécanismes de jeu de nouvelles classes internes.

Parallèlement, la richesse du jeu pourrait être accrue par des mécanismes d'alliance entre rôles, des attaques à distance, le recrutement ou le ravitaillement des troupes, différentes conditions de victoire, des zones de contrôle, etc

Enfin, il serait utile de clarifier l'usage du mécanisme d'extension, par l'identification plus précise des interfaces et classes abstraites qui devraient être particularisées pour des applications ultérieures et leur éventuel regroupement (grâce à un package rassemblant ces interfaces style `java.awt.event` par exemple).

# Annexes

## ***Annexe 1 :***

### ***La stratégie dans l'histoire***

#### **Classification des niveaux d'abstraction**

Les autorités militaires définissent de nos jours cinq niveaux d'abstraction de la gestion d'un conflit militaire ; chacun de ces niveaux est étroitement lié aux concepts immédiatement supérieur et inférieur, à l'exception du cinquième qui est partie intégrante de chacun des quatre premiers. Il importe par ailleurs de noter que les frontières ne sont pas clairement démarquées et susceptibles de varier d'une école théorique à l'autre.

- La grande stratégie –ou stratégie globale- est le niveau le plus élevé. Il se concentre essentiellement sur les aspects politiques, la sélection de buts de guerre raisonnables - qu'ils soient offensifs ou défensifs - et l'élaboration et la préservation d'alliances. Dès lors, ces décisions sont du domaine du gouvernement sous la responsabilité des cabinets, monarques, chefs de l'Etat ou ministères prenant leurs décisions sur base de l'avis de leurs chefs de service ; c'est à ce niveau qu'objectifs et priorités sont définis<sup>16</sup>.

- La stratégie a été définie par Sir Basil Liddel Hart comme « l'art de distribuer et d'appliquer des moyens militaires pour remplir les buts politiques fixés ». Dès lors, si la grande stratégie définit les objectifs d'une guerre et la forme générale qu'elle doit adopter, la stratégie détermine les méthodes par lesquelles ces buts seront atteints et ceux de l'ennemi déjoués. C'est donc le « comment », le « quand » et le « où » des opérations militaires. Actuellement, c'est le domaine des Etats-Majors nationaux, agissant conjointement avec leurs homologues d'autres pays alliés afin d'assurer une bonne coordination des efforts aux niveaux nationaux et internationaux.

- La grande tactique se préoccupe de la planification de vastes opérations dans toute leur complexité. La frontière avec la stratégie est souvent fort mince dans la mesure où ces deux strates s'interpénètrent plus que n'importe quelles autres. Le dicton du général Nathan Forrest selon lequel la stratégie signifie « allez loin le plus vite possible avec le plus de troupes<sup>17</sup> » s'applique également à ce niveau dans deux de ses aspects les plus importants : la vitesse d'exécution et la concentration de

---

<sup>16</sup> Hughes (W. P.), *"Naval Tactics and their Influence on Strategy"*.

<sup>17</sup> Très exactement : « gitting thar fustest with the mostest », l'illettrisme de ce personnage et ses idées révoltantes n'empêchant malheureusement pas qu'il ait été un bon général.

troupes. D'autres aspects comprennent le choix d'objectifs intermédiaires, l'allocation du nombre et des types corrects de troupes, la mise en œuvre d'un système de coordination efficace et la création et l'entretien d'une réserve pour exploiter la victoire ou pallier à des difficultés insoupçonnées. En général, ce type de décisions est pris à tous les échelons de l'armée à la brigade ou au régiment, voire parfois au niveau du bataillon ou de corps encore plus réduits dans le cas d'opération spéciales.

- La petite tactique gère les actions de combat et de manœuvre et implique tous les combattants depuis la simple recrue au commandant de bataillon<sup>18</sup>. L'application effective de la force armée dans le but d'atteindre des objectifs immédiats dépend donc de ce niveau d'abstraction. Bien sûr, le succès des opérations au niveau tactique est aussi vital au succès global - ou à l'échec - d'une opération que le choix de buts et plans corrects, cette chaîne n'étant que de la force de son maillon le plus faible, le plus petit détail pouvant avoir une vaste portée.

- La logistique enfin s'occupe de l'approvisionnement, du déplacement et de la mise à disposition des forces armées. Dès lors, son champ d'application est universel, couvrant la mobilisation d'économies et de populations nationales entières, l'élaboration, la création, et la production en masse d'armes et de munitions, de moyens de communication, l'organisation du transport, l'approvisionnement et la distribution géographique des dépôts destinés à fournir les forces combattantes.

Les considérations logistiques doivent être prises en compte à tous les niveaux d'abstraction du conflit : il est aussi inutile pour un dirigeant politique de demander une frappe aérienne immédiate en l'absence d'avions, armes, carburant et escortes disponibles que pour un simple lieutenant d'ordonner l'attaque d'un bunker sans grenades ou munitions. Cependant, de telles décisions doivent parfois être prises malgré l'absence de ces moyens de réalisation – ou du moins, d'une partie d'entre eux – la guerre étant fréquemment une occurrence où délais, frictions et manques sont inévitables.

La situation idéale se produit rarement, particulièrement en ce qui concerne les approvisionnements, carburants et munitions et un général trop préoccupé par ces questions n'a que peu de chances de prendre les bonnes décisions<sup>19</sup>. A l'opposé, rien ne mène plus sûrement à la défaite que l'ignorance et la négligence des nécessités logistiques à chaque niveau<sup>20</sup>. Si des différences importantes existent entre armées quant à l'organisation et la quantité de support logistique jugé

---

<sup>18</sup> Les grades étant soumis à davantage de variation suivant les époques et les lieux, nous préférons nous référer aux tailles d'unités dont les standards sont mieux établis.

<sup>19</sup> Ainsi, Rommel démontra ses grandes capacités de général en poursuivant une campagne en Afrique du Nord de 1941 à 1943 malgré une situation logistique extrêmement défavorable.

<sup>20</sup> Le succès de Montgomery à El Alamein est ainsi en partie dû à son refus d'attaquer avant que tous les éléments logistiques nécessaires ne soient en place.



nécessaire<sup>21</sup>, l'existence d'un tel support est une nécessité dans une guerre industrialisée<sup>22</sup>.

## **Classification des stratégies militaires**

Bien qu'il soit toujours dangereux de tenter une généralisation et d'imposer des structures artificielles, il n'en demeure pas moins utile de tracer les développements et tendances stratégiques majeures au cours des âges<sup>23</sup>. Dès avant l'apparition de l'écriture, les hommes se livraient à des combats, fortement liés à la chasse et qui semblent avoir essentiellement consisté en raids et attaques et fuite, ironiquement assez similaires à la forme la plus « moderne » de conflit, la guérilla. Soumis à ces attaques ayant souvent pour but l'appropriation de bétail, de reproductrices, de terres de pâture et plus tard de terres cultivables, les divers groupes se mirent en quête de refuges – souvent de grottes protégées par une barrière rocheuse – capables de résister à des agresseurs humains. Mais ces débuts de l'art de la fortification entraînèrent à leur tour des modifications dans les techniques d'agression, dans la mesure où une simple masse d'hommes mal coordonnés ne pouvait plus suffire à assurer la victoire. L'idée d'un chef coordonnant les actions avait probablement déjà trouvé un champ d'application dans le domaine de la chasse, mais ces nouvelles réalités conférèrent à ce concept une importance considérable en marquant la « naissance » du commandement militaire<sup>24</sup>.

Aux alentours de 4000 avant JC (que nous noterons dorénavant AC), les conflits primitifs laissèrent la place à ce que l'on pourrait dénommer les forces classiques de la guerre. Le besoin de prendre d'assaut des places fortes de plus en plus élaborées et bien défendues entraîna la création de véritables armées, pouvant désormais ne plus être formées que d'une partie seulement des hommes en état de porter les armes, les autres leur déléguant cette tâche. Les guerres en vinrent à se centrer autour de l'affrontement décisif, la bataille, prenant souvent les formes d'un combat juridique opposant d'importantes masses d'hommes afin de régler les différents opposant leurs deux peuples par la voie des armes. Les tactiques demeuraient simples, la plupart des batailles primitives évoluant en affrontements au corps à corps, homme contre homme. Les essais de contournement par les flancs

---

<sup>21</sup> Un soldat américain combattant au Vietnam nécessitait ainsi de douze à dix-huit hommes occupant divers rôles de support et de logistique, alors qu'un combattant Viet Min transportait lui-même la plupart de son matériel, aidé parfois d'un « boy » poussant un vélo pour la nourriture et les armes lourdes !

<sup>22</sup> Même en supposant la possibilité théorique d'une technique de ravitaillement locale (« vivre sur le pays »), le temps perdu et la désorganisation causée par ce mode de fonctionnement a pour résultat l'immobilisation d'une importante partie de l'effectif, de même que la perte de l'initiative stratégique dans un conflit traditionnel.

<sup>23</sup> Howard (M.), *The Causes of War*.

<sup>24</sup> Charnay (J.-P.), *La Stratégie*.

entraînèrent l'extension des lignes de bataille, mais au risque d'affaiblir le centre, si bien que les principes de contournement ou d'attaque massée au centre subirent de nouvelles évolutions. Les chariots conférèrent une mobilité accrue à certains groupes de guerriers, tandis que les premiers arcs et javelots se révélèrent bien plus redoutables que les anciennes sagaies propulsées par atlatl. On en vint aussi à appliquer délibérément ce que l'on pourrait qualifier de guerre psychologique, les cris de guerre initiaux étant bientôt complétés par des casques et armures fort élaborés ou des armes conçues afin d'effrayer l'ennemi, comme les feux d'artifice chinois.

Les batailles étaient dorénavant plus complexes, mais respectaient encore de vastes schémas relativement prévisibles agrémentés de variantes. Ces actions étaient tout sauf spontanées en raison du temps pris pour disposer ces formations complexes si bien qu'une telle bataille ne pouvait se dérouler qu'avec le consentement mutuel des deux commandants à moins qu'une des troupes en présence ne soit bloquée par un obstacle infranchissable l'empêchant de se retirer dans une place-forte facilement défendable. L'art de la guerre, et dès lors le test de tout commandant, consista essentiellement jusqu'à la fin du 18<sup>e</sup> siècle en la faculté de forcer au combat un adversaire contre sa volonté, au lieu et au moment désiré. Deux grandes variantes pour ce faire étaient de ravager le territoire adverse ou d'attaquer ses places fortes. Cette seconde méthode conduisit à des manœuvres de siège de plus en plus complexes où l'on divisa bientôt les troupes attaquantes en deux groupes dont l'un assaillirait le refuge tandis que l'autre repousserait les éventuelles troupes tentant de libérer la forteresse. Par ailleurs, l'on commença à mener des actions contre les lignes de ravitaillement adverses<sup>25</sup>.

La révolution française, et plus encore Napoléon, ébranlèrent ces schémas de guerre qui avaient dominé pendant des millénaires. La plus grande contribution de Bonaparte à l'art de la guerre fut la fusion de la manœuvre, de la bataille et de la poursuite en une seule séquence continue, alors que ses prédécesseurs avaient toujours établi une distinction rigide entre la marche et la bataille, adoptant des formations différentes fort coûteuses en temps pour ces deux étapes. Ainsi, Napoléon utilisa la puissance de feu supérieure qui se fit jour à son époque pour diviser son armée en détachements auto-suffisants, les corps d'armée, répartis sur plusieurs axes de progression plus ou moins parallèles, facilitant l'approvisionnement « en vivant sur le pays » et accélérant grandement les déplacements stratégiques. Il se spécialisa dans l'utilisation d'une partie de ses forces pour attirer, puis fixer l'ennemi, tandis que la plus grande part le tournait par le flanc pour couper ses lignes de communication et de retraite, obligeant le commandant ennemi à choisir entre une bataille à front inversé au lieu choisi par Napoléon ou une reddition immédiate. Le principe d'une progression dispersée avec concentration pour les batailles fut la principale raison des succès initiaux des armées napoléoniennes, leur permettant de conquérir la majeure partie de l'Europe.

Cependant, dès 1813, les adversaires de la France en savaient suffisamment long sur les méthodes napoléoniennes pour les contrecarrer efficacement : ils adoptèrent des formations d'armée similaires et ignorèrent les attaques contre leurs

---

<sup>25</sup> David (C.-P.) (dir), *Les études stratégiques. Approches et concepts*.

arrières. Le 19<sup>e</sup> siècle vit ainsi une rapide extension des fronts lors de conflits, de simples engagements locaux à des théâtres de guerre de plusieurs centaines de kilomètres de long, les lignes étant graduellement renforcées jusqu'à produire les fronts continus de la première guerre mondiale suite à l'échec des manœuvres de contournement.

Ces fronts continus, associés à un accroissement sans précédent de la puissance de feu, favorisèrent grandement la défensive par rapport à l'offensive : les fils barbelés, les mitrailleuses et les mines rendirent les assauts frontaux extrêmement coûteux en vies humaines, tandis que les autres alternatives classiques étaient rendues impossibles par la continuité même du front. Les développements technologiques successifs de gaz et des tanks se révélèrent incapables de résoudre définitivement ce problème et ce ne fut que vers 1917 -sur le front Est- et 1918 -sur le front occidental- que les Allemands découvrirent une solution partiellement efficace par l'infiltration des lignes adverses à l'aide de commandos préparant l'assaut du gros de l'infanterie. A cette date cependant, il était déjà trop tard pour que cette innovation sauve l'Allemagne du désastre. La principale raison de cet état de fait tenait en un nouveau développement de l'art de la guerre : l'effet croissant des dimensions politiques et psychologiques du conflit sur les populations civiles. Afin de réunir et d'entretenir les immenses armées nécessaires au maintien de fronts aussi étendus, les nations toutes entières furent pour la première fois totalement tournées vers l'économie de guerre. Dès lors, alors que la situation sur le front tournait à l'impasse, le côté vainqueur ne pouvait être que celui qui pourrait soutenir les tensions économiques et sociales de la guerre le plus longtemps. En 1917, ce furent des causes internes qui provoquèrent la défaite russe plus que la faiblesse de ses armées et en 1918 ce fut le tour de l'Allemagne. Le système napoléonien voyait ainsi ses postulats totalement renversés : ce n'était plus l'armée par elle-même qui était dépositaire de la volonté de combattre mais bien le peuple tout entier.

Les années d'entre-deux-guerres virent les stratégestes français, allemands et britanniques tenter de trouver une solution au blocage des fronts expérimenté durant la première guerre. Les Allemands furent les premiers à apprécier l'importance et à adapter les tactiques d'infiltration/pénétration de 1917-1918 en combinant des blindés rapides soutenus par de l'infanterie mécanisée et des bombardiers en piqué (remplaçant ici l'artillerie). Le but était de réaliser une percée limitée dans le front ennemi en évitant les points forts du dispositif ennemi, puis de poursuivre la pénétration vers l'arrière des lignes ennemies pour détruire ou désorganiser les centres de commandement et de ravitaillement adverses. Ce cette *Blitzkrieg*, basée sur les quelques divisions blindées allemandes, qui fut bien près d'apporter une victoire totale à Hitler en 1940 et 1941-1942.

Dans le même temps, une autre forme de guerre s'était développée à partir du 19<sup>e</sup> siècle : les guerres coloniales. Les premières conquêtes avaient consisté en l'envoi d'une force expéditionnaire pour écraser ou terrifier un adversaire « primitif » au moyen d'armes et de formes d'organisation supérieures ; là où la résistance s'avérait trop forte, la puissance coloniale s'appuyait sur une partie de la population locale pour « libérer » les zones en question, engendrant un état de guerre civile dont elle pouvait ensuite profiter pour étendre son influence kilomètre après kilomètre jusqu'à la totale conquête du pays. Ce système s'avéra très efficace jusqu'aux dernières années du 19<sup>e</sup> siècle, où les améliorations dans les domaines de l'éducation et de la communication de masse permirent aux dissidents de promouvoir

et répandre leurs idées non seulement pour augmenter leurs forces mais aussi pour miner la volonté de combat de la puissance coloniale par des appels aux éléments libéraux de cette dernière et, plus significativement encore, à des pays tiers. La guerre des Boers vit ainsi de nombreux appels aux Libéraux anglais et au Kaiser Guillaume. Les premiers causèrent de nombreuses frictions politiques en Grande-Bretagne mais échouèrent à détruire le désir des Anglais de gagner la guerre, comme le démontra « l'élection khaki » de 1900 ; les derniers provoquèrent des pressions internationales en faveur des Boers et leur procurèrent de l'aide matérielle. Cet épisode peut être vu comme les débuts de la guerre révolutionnaire ou de partisans, où un groupe petit mais déterminé, employant souvent des méthodes de guérilla, parvient à mobiliser l'opinion internationale en soutien de son combat. Dans ce genre d'engagements, et pour la première fois au cours de l'histoire récente, les aspects purement matériels du conflit militaire se voyaient surpassés en importance par les aspects psychologiques et idéologiques. Le résultat le plus criant en fut une rapide décolonisation après 1945, les diverses colonies acquérant l'une après l'autre l'indépendance, et ce souvent après avoir perdu le conflit en termes purement militaires. Le but de ces mouvements était de tenir jusqu'à ce que les pressions internationales et l'épuisement de la métropole l'obligent à reconnaître l'indépendance de l'ancienne colonie<sup>26</sup>.

Les aspects stratégiques induits par la menace nucléaire, parmi lesquels l'impossibilité d'un conflit conventionnel, ont quant à eux tellement révolutionné l'art de la guerre traditionnel que la diplomatie « grise » et les conflits indirects ont paradoxalement ramené les engagements à une échelle plus réduite tout en étendant leur portée au monde entier<sup>27</sup>.

### **Les sept manœuvres classiques de la guerre :**

Depuis des temps immémoriaux, les manœuvres effectuées par les divers commandants peuvent être classifiées en 7 grandes catégories. L'apparition de nouvelles armes et de développements technologiques n'a pas radicalement altéré ce schéma, mais plutôt les méthodes de sa mise en œuvre. Cependant, l'époque contemporaine a vu la naissance d'une dérivation de l'approche indirecte qui en viendra sans doute à être considérée comme une huitième manœuvre à part entière.

#### *1-Pénétration du centre*

Sans doute la plus ancienne manœuvre, remontant aux premières armées. Les premières batailles semblent avoir pris la forme d'une avance au contact, couverts par des volées de projectiles, pour se subdiviser à ce moment en une multitude d'engagements individuels jusqu'à ce qu'une des armées s'enfuit. Plus tard, un commandant découvrit l'importance de conserver une part de ses troupes en réserve

---

<sup>26</sup> Luttwak (E. N.), *Strategy and History*.

<sup>27</sup> Charnay (J.-P.), *Critique de la stratégie*.

pour les envoyer au moment opportun briser le centre adverse et vouer l'armée adverse à la fuite ou la destruction.

La plupart de ces engagements opposaient presque uniquement des fantassins ; ainsi, en 1066, tant les Anglo-Saxons que les Danois choisirent de combattre démontés à Stamford Bridge. Cependant, la cavalerie joua également un rôle vital dans plusieurs batailles impliquant ce modèle jusqu'à une époque récente : le duc de Marlborough gagna trois de ses quatre grandes batailles en perçant le centre adverse à l'aide d'une force de cavalerie gardée en réserve. Lors de la seconde guerre mondiale, la percée allemande de Sedan les 14 et 15 mai 1940, ou le succès de Montgomery lors de la troisième phase de la bataille d'El Alamein vers la fin octobre 1942, en fournissent des exemples parlants. Cette méthode fut souvent appliquée lorsque existait une continuité du front ; parmi ses avantages, celui de pouvoir encercler de larges portions des forces adverses ou de poursuivre cette percée vers les bases arrières ennemies. Les désavantages éventuels incluent l'affaiblissement excessif des flancs nécessaires à l'établissement d'une force de frappe suffisante au centre et la possibilité qu'une percée ne mène pas à une pénétration en profondeur des lignes ennemies si ce dernier achemine rapidement des renforts vers le secteur enfoncé, comme les batailles de Cambrai et de la Somme l'ont montré.

## *2-Enveloppement d'un flanc*

Comme les premières armées se dotaient de chariots, d'éléphants ou de cavalerie, leur puissance manœuvrière s'accrut grandement. Un général talentueux, tandis qu'il maintenait les forces adverses grâce à son infanterie, pouvait lancer ses forces montées en un vaste mouvement contournant l'extrémité de la ligne adverse. Si cette manœuvre était menée avec succès, cela permettait le retournement progressif de toute la ligne de bataille adverse, ou tout au moins de nuire grandement à sa cohésion ; un opposant compétent pouvait néanmoins s'apercevoir de ce danger et le contrer à l'aide de ses propres forces montées.

Cette méthode bénéficia d'une grande popularité depuis l'Antiquité à nos jours ; ainsi, Alexandre l'employa à Gaugamèles en 331 AC pour écraser l'armée perse, tandis qu'en mai 1942, Erwin Rommel réitéra la même manœuvre pour contourner le flanc Sud de la huitième armée britannique chargée de tenir la ligne de Gazala, parvenant ainsi à capturer Tobrouk et faire reculer les Britanniques jusqu'à Alexandrie.

Si l'avantage de cette manœuvre apparaît dès lors évident, son principal danger est l'affaiblissement du centre de l'armée pour masser une force suffisante sur l'aile, permettant à un adversaire décidé de retourner la situation, ce qui advint aux Russes et Autrichiens à Austerlitz en décembre 1805.

## *3-Enveloppement des deux flancs*

Si elle est couronnée de succès, cette manœuvre peut mener à l'anéantissement total de l'ennemi. Pour ce faire, il faut ou une supériorité numérique écrasante, ou de grands talents de général puisque tout adversaire fera tout pour éviter cette situation.

Les grands exemples classiques sont Cannes (216 AC), où Hannibal et son armée de 50 000 hommes surpassèrent tellement le consul Varron et ses 80 000 troupes que seuls quelques milliers purent s'enfuir vers Rome après la plus grande défaite militaire de l'histoire romaine. En 1914, le double enveloppement mené par Hindenburg et von François détruisit deux armées russes à la bataille de Tannenberg en Prusse Orientale. Trente-sept ans plus tard, la même manœuvre permit à Kleist et Guderian de remporter une écrasante victoire à Kiev en encerclant le groupe d'armée du maréchal Budenny, capturant ainsi plus de 500 000 Russes. Deux ans plus tard, le même sort fut réservé aux 200 000 Allemands de la septième armée lors de la bataille de la falaise de Gap en Normandie.

Si cette méthode offre la perspective d'une destruction totale de l'ennemi, son principal danger réside dans la sur-extension des troupes pour compléter l'encercllement, permettant à l'ennemi d'effectuer une percée et même de retourner la situation si les lignes sont trop minces, comme cela arriva à Kohima-Imphal en Birmanie en 1944.

#### *4-Attaque en ordre oblique*

Au cours de cette manœuvre, un commandant masse graduellement des forces de plus en plus importantes sur une aile jusqu'à ce que la ligne ennemie cède, tandis que les faibles forces présentes sur l'autre aile doivent attirer l'attention de l'ennemi et éviter le transfert de ses réserves vers son flanc menacé. Utilisée sans doute pour la première fois en 371 AC à la bataille de Leuctres par le stratège thébain Epaminondas contre les Spartiates, Frédéric le Grand de Prusse en fit un usage intensif à une époque plus récente. Ainsi, en 1757, il écrasa le flanc droit autrichien sous une énorme poussée de ses troupes. Sous une forme plus élaborée, cette manœuvre fut employée par Napoléon lors des batailles de Castiglione (1796) et Bautzen (1813). Par essence, elle implique la constitution d'une masse de rupture dans le plus grand secret en face du secteur visé, si bien qu'elle peut être comparée à la première manœuvre sur ce point.

Employée correctement, elle peut permettre la dislocation de la ligne ennemie ; si les choses tournent mal cependant, l'attaquant court le risque d'une contre-attaque sur son flanc le plus dégarni.

#### *5-Retraite simulée*

Cette manœuvre a souvent été associée au double enveloppement : son but principal est d'induire l'adversaire à abandonner une solide position défensive ou à se précipiter dans une embuscade, par la perspective d'une victoire facile suite à la fuite des troupes au contact. Cannes en est un exemple évident, puisque Hannibal ordonna à son centre de faire retraite devant les troupes de Varron jusqu'à ce que celles-ci soient désorganisées et massées dans un espace bien trop exigü pour manœuvrer, menant au massacre que l'on sait. En 1066, Guillaume le Conquérant tenta de la même manière le fyrd anglo-saxon indiscipliné lors de la bataille d'Hastings ; dès que cette milice s'élança en poursuite de Senlac Hill, elle fut taillée en pièce par les Normands, leur permettant ensuite d'encercler le mur de boucliers des thegns d'Harold.

Une variante fut employée par Wellington devant Salamanque en 1812 : l'apparente retraite des forces alliées vers le Portugal poussa le maréchal Marmont à disperser ses troupes dans l'espoir de couper la route du Portugal, permettant à Wellington de prendre l'armée française de flanc, écrasant toutes les divisions sauf celles d'arrière garde. Au cours des siècles précédents, cela avait aussi constitué la tactique favorite des armées turques, vastes mais peu organisées. Elles feignirent la fuite plusieurs fois lors de la campagne de Vienne afin de pousser les troupes autrichiennes à quitter leurs positions défensives dans l'espoir de butin, faisant dire au prince Eugène : « Je crains l'armée turque moins que son camp ». A l'époque contemporaine, cette même manœuvre fut employée par le général O'Connor lorsqu'il se replia de la frontière libyenne en 1940 avant de contre-attaquer et de disperser la dixième armée italienne, désorganisée par cette poursuite. Rommel fit de même en janvier 1942, lorsque après une longue retraite depuis Tobrouk, il se retourna à Agheila et infligea une série de défaites à la huitième armée anglaise qui se trouva contrainte d'abandonner Benghazi et la plus grande part de la Cyrénaïque.

Les dangers associés à cette méthode sont néanmoins considérables : lorsque des troupes commencent à faire retraite, même de façon planifiée, la différence entre retraite et fuite peut être fort mince. Le moral des troupes dont on attend cette manœuvre doit être fort élevé ou, comme à Cannes, la présence d'un obstacle infranchissable doit obliger les troupes à faire face ou mourir. Si cette attaque est menée par un commandant calme et compétent, ses effets peuvent être dévastateurs, tant matériellement que psychologiquement.

#### *6-Attaque depuis une position défensive*

Napoléon écrivait : « Tout l'art de la guerre consiste en une défense bien raisonnée et extrêmement circonspecte, suivie d'une attaque rapide et audacieuse. Dans cette méthode, une armée prépare une forte position défensive puis pousse l'ennemi à l'attaquer. Lorsque celui-ci a épuisé ses troupes en vaines attaques contre le périmètre défensif, la garnison fait une sortie pour disloquer l'armée affaiblie<sup>28</sup>. Cette manœuvre connaît de nombreuses variantes : dans sa forme la plus simple, l'ennemi est poussé à attaquer une position fortifiée, comme à Bannockburn, Crécy, Poitiers et Azincourt, où des rangées de pieux et des fosses à piques réduisirent à néant les charges de cavalerie lourde adverses tout en fournissant la couverture nécessaire aux piquiers et archers pour décimer les lentes troupes d'infanterie envoyées par la suite. Sous une autre forme, le commandant oblige son adversaire au combat en assiégeant une ville importante, puis en se laissant assiéger à son tour avant de passer à l'offensive ; cette manœuvre fut notamment tentée –et réussie– tant par Jules César lors du siège d'Alésia (52 AC) que par le prince Eugène à Belgrade (1717).

Un exemple raffiné d'une troisième variante peut être trouvé dans l'utilisation faite par Wellington de la ligne de Torres Vedras au Portugal afin de confondre le maréchal Masséna. Après avoir amené Masséna devant cette ligne fortifiée par une

---

<sup>28</sup> Booth (K.), Herring (E.), *Strategic Studies*.

longue retraite, Wellington fit face, abrité derrière une triple ligne de fortifications, usant de la technique de la terre brûlée et d'actions de guérilla pour affaiblir son ennemi avant de se lancer à sa poursuite. Plus récemment, la bataille de Kohima-Imphal révéla les talents de commandant de Slim et la qualité de sa quatorzième armée. Son quatrième corps fut chargé de pousser Mutaguchi à attaquer les « boîtes » discrètement préparées le long de la route franchissant les monts Assam en direction de l'Inde. Ensuite, grâce en partie à la maîtrise aérienne des alliés, le général Scoones résista à tous les assauts jusqu'à l'épuisement des troupes japonaises, puis contre-attaqua en juin, repoussant les troupes adverses jusqu'à la rivière Chindwin lors de la fameuse « avance de la mousson ».

De telles tentatives peuvent cependant avoir des conséquences catastrophiques. Pour Stalingrad, en 1942, Hitler crut les assertions de Goering sur la capacité de la Luftwaffe à assurer indéfiniment le ravitaillement aérien de la sixième armée de von Paulus, et l'écroulement du ravitaillement aérien fut la première cause de cette écrasante défaite. Quelques années plus tard, en 1954, les Français établirent délibérément une position défensive à Dien Bien Phu, loin derrière les lignes adverses et à proximité d'une importante route de ravitaillement Viet Min, désirant forcer Giap à la confrontation. Si cette partie du plan se déroula comme prévu, la bataille qui s'ensuivit coûta l'Indochine à la France, suite tant à l'insuffisance du ravitaillement aérien qu'à une sous-estimation de l'ennemi et particulièrement de son artillerie. Il est intéressant de constater que Giap douta à un moment du succès et que lorsque l'occasion lui fut présentée quelques années plus tard d'infliger une telle défaite aux Américains à Khe-Sanh en 1969, il préféra se retirer dès les premiers signes annonciateurs de troubles.

Efficacement employée, cette méthode peut conduire à des résultats décisifs ; elle peut aussi néanmoins conduire à de dangereuses doctrines défensives, comme dans le cas de la ligne Maginot pour les Français en 1940. Pour fonctionner, cette méthode nécessite un ennemi désireux d'attaquer, sans quoi les troupes fortifiées peuvent attendre indéfiniment, voire se faire attaquer d'une direction inattendue, comme ce fut le cas pour Singapour au début de 1942, lorsque les Japonais l'attaquèrent par voie de terre.

### *7-Approche indirecte*

Ce concept implique la distraction de l'ennemi à l'aide d'une force secondaire, tandis que la force principale effectue une marche destinée à envelopper le flanc et l'arrière de l'ennemi au niveau stratégique. L'effet d'une telle manœuvre, si elle est menée avec succès, est d'interrompre les ravitaillements ennemis, nuire à son moral et l'obliger à accepter une bataille à front inversé<sup>29</sup> ne pouvant mener qu'au désastre pour le camp vaincu en l'absence de ligne de retraite. Les critères de rapidité et de surprise sont évidemment vitaux au succès de l'entreprise, cette méthode pouvant être considérée comme la suprême manifestation de l'art de la guerre offensif jusque dans les années 40'. Elle fut tentée à de multiples reprises par les plus grands

---

<sup>29</sup> C'est-à-dire où chaque camp fait face à ses bases.



commandants de l'Antiquité, Alexandre, Hannibal ou César, mais rarement pleinement réussie en raison du manque de mobilité des troupes de l'époque. Elle parvint presque à la perfection aux mains de Napoléon, soutenant pas moins de trente de ses principales manœuvres entre 1796 et 1812, dont le balayage d'octobre 1805 afin d'enfermer le général Mack dans Ulm. Elle fut exemplairement mise en œuvre par Robert E. Lee contre Pope lors de la seconde campagne de Bull Run en 1862.

Cette manœuvre peut bien sûr être associée avec d'autres, développant alors tout son potentiel : ainsi, en 1940, la conversion par les troupes allemandes d'une percée stratégique contre le centre allié dans la région de Sedan en un vaste enveloppement tandis que les Panzers fonçaient vers la côte de la Manche, encercla 330 000 Britanniques dans la région de Dunkerke. S'ils furent finalement sauvés, tout leur équipement dut être abandonné. Une application encore plus réussie de ce principe se déroula en 1945 en Birmanie, lorsque Slim attira l'attention de Kimura à Mandalay tandis qu'il préparait le gros de ses troupes à une vaste offensive bien plus au Sud contre le centre de ravitaillement japonais de Meiktina. Le succès de cette offensive condamna la défense japonaise de la Birmanie<sup>30</sup>.

L'avantage de ce système est la promesse de résultats décisifs et sa capacité à forcer le combat même face à un adversaire voulant s'y refuser, puisque les conséquences d'un repli équivaldraient à celles d'une défaite. Il nécessite néanmoins une forte supériorité des forces et un grand sens des délais puisqu'une application trop désinvolte peut conduire à des résultats catastrophiques en permettant à un adversaire alerte de contrer la menace –comme ce fut le cas pour Napoléon en 1813 et 1814- en écrasant successivement les différentes tentatives de débordement.

#### *8- Application moderne de l'approche indirecte : les guérillas et guerres révolutionnaires*

Les plus importantes leçons à retenir de ces sept manœuvres sont l'importance de gagner et de conserver l'initiative et de déstabiliser l'adversaire sans lui permettre d'en faire autant. Ces principes demeurent valables jusque dans les formes les plus modernes de conflits, même si leurs formes ont été profondément altérées depuis 1945. En effet, depuis cette date, de nombreuses « guerres » ont été provoquées par la volonté d'indépendance d'anciennes colonies, puis parfois les luttes internes pour le pouvoir, toutes employant des méthodes de guérilla à plus ou moins grande échelle. Par sa définition même, la guérilla est basée sur des moyens indirects – terreur, raids, sabotages, propagande- mais son utilisation offensive en tant que moyen de gagner une prééminence politique et une supériorité psychologique est un développement relativement récent. La guerre de partisans fut définie par Napoléon comme « une guerre sans front », tandis que Che Guevara présentait le but de la guérilla comme celui « d'être partout mais nulle part ». Jusqu'en 1945, les tactiques de guérilla n'étaient utilisées qu'en dernier ressort, le but principal étant d'éviter une

---

<sup>30</sup> Phélizon (J.-F.), *L'Action stratégique*.

défaite totale en formant un noyau de résistance à l'envahisseur, comme en Espagne (1808-14), Afrique du Sud (1900-01) ou Yougoslavie (1941-44). Depuis cependant, ce concept s'est modifié pour couvrir des actions offensives mais de faible proportions<sup>31</sup>.

Une partie de l'originalité de ce concept tient à la faible importance relative des succès militaires conventionnels puisque même en perdant le conflit militaire, les guérilleros peuvent sortir vainqueurs du conflit politique. La valeur accordée aux victoires militaires a presque été renversée puisque les Français gagnèrent la guerre d'Algérie mais durent se retirer par la suite, tandis que le même sort échu aux Anglais après qu'ils aient écrasé la révolte Mau Mau au Kenya. La seule solution à long-terme est la prévention des troubles économiques, sociaux et politiques par une prompt action avant d'atteindre le stade de la confrontation. Toutes les autres méthodes sont au mieux des palliatifs, au pire des aggravations de la situation. Lorsqu'une action militaire est entreprise, la priorité semble devoir être de séparer les guérilleros de leurs soutiens au sein de la population, en améliorant les conditions de vie de la population et en infiltrant les réseaux d'aide aux rebelles ; c'est seulement une fois ces objectifs pleinement atteints que la traque des groupes actifs peut avoir la moindre chance de succès, comme le démontra le succès local et temporaire des forces britanniques à Malaga (1948-60). Il semble donc qu'une huitième manœuvre –favorisant le groupe le plus faible mais le plus motivé par rapport au plus fort- doive dorénavant être reconnue<sup>32</sup>.

## **Annexe 2 :**

### ***La stratégie japonaise lors de la guerre du Pacifique***

Dans la mesure où le côté japonais du conflit s'avère le moins connu, il nous semble intéressant d'analyser brièvement le contexte et l'enchaînement des décisions stratégiques japonaises.

Le phénomène de flux et de reflux est le caractère le plus frappant de la guerre du Pacifique. En 1941, le Japon a littéralement explosé et, dans un flot rapide de victoires, a occupé de vastes régions dans le Sud-Est asiatique et dans le Pacifique ; cependant, au bout de sept mois, ce flot a été arrêté dans la mer de Corail ainsi qu'à Midway pour se retirer de Guadalcanal au bout de quatorze mois. Avec une force sans cesse grandissante et une grande rapidité, le flot américain submergea alors le Pacifique, balayant sans difficultés stratégiques toutes les régions encore occupées du Pacifique jusqu'en 1945 où seul le Japon et ses dépendances insulaires demeuraient sous le contrôle plein et entier de l'armée nipponne. Le caractère

---

<sup>31</sup> Cimbala (S. J.), *"Military Strategy"*.

<sup>32</sup> Luttwak (E. N.), *Le Paradoxe de la stratégie*.

absolu de l'avance et du recul japonais comparé au va-et-vient de la campagne d'Afrique du Nord incite à se limiter à une vision simple et largement répandue de la lutte en Extrême-Orient : les succès initiaux des Japonais seraient dus à l'impréparation des alliés soumis à des attaques brillamment exécutées. Les Japonais, suite à ces avances spectaculaires – et inattendues -, se seraient étendus outre-mesure et se seraient révélés incapables de contenir l'assaut des Etats-Unis dès le passage de ceux-ci à l'économie de guerre. Cette généralisation, bien qu'intéressante et assez juste, est en l'occurrence bien trop simpliste pour résister à une analyse plus fine. Son principal défaut est d'ignorer les décisions stratégiques des deux adversaires principaux et leur énorme influence sur les batailles et l'orientation générale de la guerre. En d'autres termes, ce flot ne se porta pas en avant pour ensuite reculer uniquement à cause d'une disproportion entre les économies mises en présence, mais aussi du fait des erreurs stratégiques et tactiques commises par l'un et l'autre adversaire dont l'absence aurait pu changer le cours de la guerre. C'est dans cette perspective que nous nous proposons d'étudier les décisions japonaises, dans la mesure où elles sont souvent les plus méconnues.

Dans le système de commandement japonais, l'organisme qui avait pour mission principale de prendre les décisions stratégiques était l'Etat-Major général impérial (c'est-à-dire les Etats-Majors généraux de l'Armée et de la Marine sous le commandement suprême de l'Empereur) ; toutefois, cet organisme était influencé par les suggestions et critiques venant tant de petits groupes que de l'amiral commandant en chef à la Flotte. Ce système n'avait peut-être en soi rien d'exceptionnel, mais l'emprise du pouvoir militaire sur le pouvoir civil était son trait le plus marquant. Le Cabinet, qui n'était pas responsable devant la Diète mais seulement devant l'Empereur, était dominé par les Ministres de la Guerre et de la Marine, sans lesquels cette institution ne pouvait exister. Ces deux ministres étaient eux-mêmes choisis par les forces armées dont ils étaient les porte-paroles ; de plus, suite à cette influence, le premier ministre était souvent choisi parmi les militaires. De toutes manières, les Chefs d'Etat-Major avaient accès directement à l'Empereur sans même avoir à en informer le Cabinet. Enfin, comme l'Empereur n'avait pas à assumer la responsabilité de la décision, il était seulement censé entériner les décisions du Cabinet et du Quartier général impérial, permettant au clan militariste d'entraîner le gouvernement dans n'importe quelle action aventureuse et le pays vers la guerre alors que ces organes auraient dû limiter leurs compétences à des questions de stratégie ou de tactique opérationnelles<sup>33</sup>.

Les objectifs du gouvernement japonais, calqués sur ceux de l'Armée et de la Marine, étaient résolument expansionnistes et très ambitieux ; en tant que nation militariste invaincue et la plus avancée industriellement de tout l'Extrême-Orient, le Japon fut naturellement tenté de suivre l'exemple des nations européennes en jouant un rôle impérialiste. Avec un appétit aiguisé par ses gains de la guerre 1914-1918 et son peuple déçu par la crise économique des années 1930, ainsi que par l'attitude réservée et sévère des Anglo-Saxons, les conditions étaient réunies pour que les militaires entraînent leur pays dans des guerres de conquêtes territoriales sur le continent asiatique. La Mandchourie absorbée et transformée en Etat fantoche,

---

<sup>33</sup> Bauer (E.), *Histoire Controversée de la Deuxième Guerre mondiale*, volume 5, pp. 234-258.

l'expansion continua à un rythme rapide en Chine même à partir de 1937. Cette campagne ne représentait déjà à l'époque qu'une faible partie d'un plus vaste ensemble qui visait à la création d'une « Sphère de prospérité communautaire du Grand Extrême Orient Asiatique », zone de domination économique et politique du Japon s'étendant depuis la Corée jusqu'à la Birmanie et du Yang Tse Kiang au centre du Pacifique.

Une solution de rechange à cette politique était l'expansion vers le Nord ; après tout, la Russie avait été l'ennemie du Japon au début du siècle et les victoires incontestables remportées sur la flotte russe avaient garanti au Japon le contrôle de la Mandchourie et des régions limitrophes. Cependant, ce plan soulevait de nombreuses objections : d'abord, le *statu quo* était souhaité car le Japon n'avait que peu à gagner de cette expansion (la steppe sibérienne) et beaucoup à perdre (la Mandchourie, la Corée, la Chine du Nord) en s'engageant dans une lutte sauvage contre les Soviétiques. En second lieu, la cuisante défaite qui leur avait été infligée à l'occasion de l'incident de Nomonhan en 1939 leur firent douter de leur victoire dans une telle lutte, alors que la Sibirie ne pouvait en rien se comparer avec les Indes ou la Chine. Finalement, seule l'expansion vers le Sud pouvait résoudre le problème de l'enlèvement graduel en Chine<sup>34</sup>. En conséquence, les Japonais conclurent un pacte de neutralité avec la Russie en avril 1941. Rassuré par cet accord sur son flanc Nord, et plus tard par l'attaque d'Hitler contre la Russie, le Japon était libre de préparer son offensive vers le Sud. Mais les décideurs militaires ne se sentirent pas suffisamment protégés à cet égard pour rappeler leur garnison de Mandchourie, forte de treize à quinze divisions<sup>35</sup>, qui demeura en place durant toute la durée du conflit, alors que ces troupes firent cruellement défaut par la suite sur d'autres théâtres d'opérations. Ce fut une des erreurs initiales de l'Etat-Major japonais que d'immobiliser une si grande masse d'hommes en vue d'une guerre hypothétique.

A la surprise complète du Japon, aucune victoire décisive ne pouvait être entrevue en Chine et, à partir de 1937, un nombre toujours croissant de troupes fut immobilisé par la poursuite de la lutte contre le gouvernement nationaliste chinois. Plus les complications et les frustrations apparaissaient chez les militaires, plus était vive la tentation de couper par le Sud les lignes de ravitaillement de la Chine et parallèlement plus grand était le risque d'entrer en conflit avec les puissances occidentales, en particulier avec les Etats-Unis, qui considéraient comme inquiétants les défis japonais. Ainsi, la décision prise par le Japon d'entrer en guerre en 1941 n'était pas seulement une décision subite et irréfléchie mais la plus importante d'une série d'escalades militaires.

En juillet 1941, l'occupation japonaise de l'Indochine, qui était un grand pas en avant dans l'encerclement de la Chine, entraîna de la part des Américains, des Britanniques et des Hollandais une action de représailles. En gelant les avoirs du Japon, ces puissances coupèrent son approvisionnement en hydrocarbures, vital pour lui, et surtout pour ses opérations en Chine, lesquelles risquaient de s'arrêter

---

<sup>34</sup> Keegan (J.), *Atlas de la Seconde Guerre mondiale*, pp. 111-114.

<sup>35</sup> Environ 700 000 hommes.

faute de carburant. A moins de renoncer à leur politique, la seule solution qui restait aux Japonais était de s'emparer des riches gisements pétrolifères des Indes Orientales. Une conférence impériale, composée de l'Empereur, des représentants des forces armées et des membres importants du gouvernement, avait décidé le 2 juillet de poursuivre l'offensive vers le Sud, même si cela devait entraîner un conflit avec les puissances occidentales. Au cours d'une conférence qui eut lieu dans les premiers jours de septembre, la décision fut prise d'entrer en guerre si le Japon ne pouvait atteindre ses objectifs minima par des pourparlers avec les Anglo-Saxons. Mais étant donné que ces négociations n'aboutissaient pas et comme les chefs des Etats-Majors de l'Armée de terre et de la Marine (Général Sugiyama et Amiral Nagano) se montraient de plus en plus inquiets du fait de la diminution de l'approvisionnement en matériel de guerre, le Ministre de la Guerre Tojo contraignit le prince Konoye à démissionner pour prendre sa place de Premier Ministre. Des lors, les généraux prirent les affaires en main et une conférence impériale décida au début de novembre l'entrée en guerre, décision confirmée par une dernière rencontre le 1<sup>er</sup> décembre suivant.

Pendant ce temps, les plans de guerre japonais furent discutés ; à mesure que les discussions avançaient, les zones susceptibles d'être attaquées s'agrandissaient considérablement en invoquant la pure logique militaire. En fait, le Japon avait besoin des champs pétrolifères de Java, Bornéo et Sumatra ainsi que de la possession de la route de Birmanie si la conquête de la Chine devait être achevée. Puisque les forces des Etats-Unis stationnées aux Philippines et celles des Britanniques installées en Malaisie s'opposeraient vraisemblablement à de telles offensives - il n'était guère plausible que les puissances ayant établi l'embargo contre le Japon acceptent de le voir levé par la force - , les Japonais devaient aussi s'emparer de ces zones. Mais alors, rien ne garantissait la non-intervention de la flotte américaine basée à Pearl Harbour ; le Commandant en Chef de la flotte combinée, Yamamoto, estimant le danger réel, insista pour que cette menace soit également éliminée.

Le succès de la stratégie japonaise exigeait une action rapide et concertée afin de bénéficier de l'effet de surprise, tout autant que l'utilisation de la puissance aérienne à une vaste échelle. Pour une opération aussi complexe et ambitieuse, les Japonais y affectèrent des troupes peu nombreuses - onze divisions par comparaison aux treize stationnées en Mandchourie et aux vingt-deux en Chine – mais près de 1200 avions et la plus grande partie de la Marine. Les trois principales phases devaient être les suivantes<sup>36</sup> :

- la flotte américaine de Pearl Harbour serait neutralisée tandis que des offensives contre la Thaïlande, la Malaisie et Hong Kong seraient lancées par l'Armée ; ensuite des raids aériens seraient entrepris contre les Philippines, Guam et Wake, précédant de peu des débarquements aux Philippines et à Bornéo.

---

<sup>36</sup> Dull (P. S.), A Battle History of the Imperial Japanese Navy 1941-1945, pp. 62-84.

- Ces régions une fois occupées, les troupes japonaises poursuivraient leur offensive en s'emparant de la Malaisie, de Singapour et des bases aériennes du sud de la Birmanie, puis de l'archipel des Bismark et des points stratégiques des Indes Néerlandaises.
- Une fois l'occupation de Java et de Sumatra achevée, les Japonais se porteraient vers les Indes et s'empareraient des îles Andaman et Nicobar dans le golfe du Bengale. Ils espéraient atteindre tous ces objectifs dans un délai d'environ six mois.

Tout à la planification détaillée des opérations, les Japonais n'attachèrent que peu d'importance à la question de savoir ce qui allait se passer une fois la victoire acquise. D'une manière générale, ils se satisfaisaient de l'idée que l'Amiral Morison exprime en parlant de « *Festung Nipona* » - c'est-à-dire la création d'une zone circulaire de défense allant des Aléoutiennes à la Birmanie qui serait à l'abri des tentatives américaines de pénétration et ainsi obligerait les Etats-Unis à reconnaître les conquêtes du Japon. Cependant, peu de préparatifs de longue haleine avaient été faits pour renforcer une telle défense et peu de personnes à Tokyo se posaient la question de savoir si oui ou non les Etats-Unis pouvaient être forcés d'accepter une si vaste extension de la zone de contrôle japonaise et ainsi subir l'humiliation d'une défaite. En fait, mis à part Yamamoto et quelques autres, peu de gens s'interrogeaient sur la portée d'une décision d'une grande importance stratégique : s'attaquer aux Etats-Unis eux-mêmes. Valait-il la peine de provoquer à coup sûr une réplique armée d'une puissance dont l'incroyable potentiel économique tourné vers l'industrie de guerre lui permettrait d'être victorieuse dans une guerre de longue durée ? Entièrement occupés par la recherche de la gloire militaire, par leur lutte en Chine et par leur réaction contre l'embargo du pétrole, les décideurs firent fi de ces préoccupations en se lançant dans la guerre<sup>37</sup>.

L'assaut japonais fut totalement couronné de succès : dans le délai de six mois prévu, tous les objectifs avaient été atteints à moindre frais, isolant la Chine et anéantissant au cours de cette offensive les forces armées alliées. Jamais auparavant une zone aussi étendue n'avait été conquise en un temps aussi court, conjonction d'un armement efficace, du courage des soldats japonais et d'une excellente planification. Les chefs militaires japonais eux-mêmes semblent avoir été surpris par l'ampleur de leurs succès. Dans l'euphorie de la victoire, l'idée de consolider leurs gains fut négligée en faveur d'une nouvelle avance, même si la nature exacte de cette dernière restait âprement débattue. Plusieurs groupes d'influence proposèrent des plans diamétralement opposés et une certaine confusion s'installa qui fit que la planification stratégique, par ailleurs excellente, présenta les premiers signes de dérapages ; des fautes furent commises et des occasions inespérées laissées inexploitées.

En fait, c'est l'Armée qui doit ici être considérée comme principal responsable de ces attermoissements - d'autant plus qu'elle a toujours été l'autorité principale en matière de stratégie - : en effet, depuis des années, l'Etat-Major général portait ses

---

<sup>37</sup> Purnell (C.) (dir), *La Deuxième Guerre mondiale*, pp. 235-247.

vues sur la guerre en Chine et sur la défense de la Mandchourie et, outre les problèmes issus de la possession de territoires aussi exotiques que Singapour ou Rabaul, il commença à craindre qu'une prochaine expansion ne pousse à puiser dans les forces engagées sur le continent. La suggestion de l'Etat-Major de la Marine de s'emparer de Ceylan fut ainsi rejetée par l'Armée qui ne désirait pas se voir amputée d'autres forces. Au cours de cette phase de la guerre, les Britanniques, toujours sous le coup de leur succession rapide de défaites considéraient une attaque de Ceylan et du Sud des Indes par les porte-avions japonais - alors que la faiblesse des défenses aériennes et navales de ces régions avait déjà été démontrée – comme éminemment sérieuse, sinon décisive. En effet, la flotte de porte-avions japonaise, une fois basée à Ceylan, aurait non seulement coupé les communications avec les Indes mais aurait pu s'attaquer à celles d'Egypte et du Golfe Persique - entravant également la concentration des troupes en vue de la bataille d'El Alamein - . Une jonction entre les Allemands et les Japonais au Moyen-Orient était loin d'être impossible durant cette phase de la guerre mais l'Armée rejeta une telle éventualité<sup>38</sup>.

La Marine avait d'autres plans qui étaient loin de faire l'unanimité, à savoir les opérations contre l'Australie et Hawaii, les deux bases probables d'une contre-attaque alliée. L'invasion de ces deux zones impliquait l'emploi d'un nombre trop élevé de troupes aux yeux de l'Armée, mais l'action subtile de l'Etat-Major général de la Marine leur fit donner leur approbation aux opérations suivantes en avril et mai 1942 :

- une offensive dirigée sur Tulagi dans les Salomon et sur Port Moresby en Nouvelle Guinée ;
- l'occupation des îles Midway et des Aléoutiennes avec la neutralisation de la flotte du Pacifique qui venait de lancer le raid de Doolittle sur Tokyo ;
- d'autres opérations en direction des Nouvelles-Hébrides et de Samoa afin de couper les lignes de communication entre les Etats-Unis et l'Australie.

La première opération conduisit à la bataille de la Mer de Corail, la seconde à l'affrontement de Midway et la troisième aux combats de Guadalcanal. Les trois opérations furent des échecs, les deux dernières si désastreuses que le Japon perdit toute initiative dans la guerre du Pacifique et ne put jamais la regagner. Elles furent perdues toutes trois tant du fait d'erreurs de planification du Grand Quartier-Général impérial que de fautes tactiques du commandement que les Américains ne se firent pas faute d'exploiter au mieux. L'erreur fondamentale des Japonais a été leur présomption excessive à ce stade, conséquence de leurs premières victoires écrasantes à l'aide de forces peu nombreuses. Pour pouvoir gagner la bataille de Mer de Corail, il leur aurait fallu faire intervenir six de leurs porte-avions d'escadre et

---

<sup>38</sup> Masson (P.), *Précis d'histoire de la Seconde Guerre mondiale*, pp. 182-187.

non deux, même si cela risquait de retarder l'attaque contre Midway<sup>39</sup>. L'avantage qu'apportait une telle victoire - le contrôle de la Nouvelle Guinée et des Salomon - en valait la peine.

Lors de la bataille de Midway, les Japonais commirent un nombre d'erreurs extraordinaire. A nouveau, l'armée fut la première responsable de la situation en se montrant extrêmement parcimonieuse à propos de l'invasion d'Hawaii. On peut raisonnablement supposer que l'attaque de cet archipel à l'aide de la flotte et d'environ cinq divisions aurait été le coup le plus terrible à la faculté de sursaut américaine dans le Pacifique. Non seulement ils auraient coupé les communications avec l'Australie et se seraient emparés d'une base pouvant abriter leur flotte, mais ils auraient forcé les Américains à reporter leur ligne de défense à près de 5000 kilomètres de là, près de la côte californienne et obligé les porte-avions américains au repli pour protéger le territoire national s'ils survivaient à l'attaque<sup>40</sup>. C'est ensuite l'Etat-Major général de la Marine et la flotte combinée qui ont commis une maladresse en subordonnant, dans un plan soigneusement élaboré, les mouvements de toutes les flottes à celui de la flotte d'invasion de Midway, fort lente. Ainsi, l'objectif majeur à ce stade du conflit - la destruction des porte-avions et la conquête d'une position stratégique - fut sacrifié au profit d'une bataille pour une île à l'utilité discutable, ce qui fit finalement qu'aucun objectif ne fut atteint. Tactiquement, l'opération fut bâclée du commencement à la fin et même Yamamoto fut incapable de redresser la situation puisqu'il s'était de lui-même séparé de sa force principale, ses porte-avions. Le Japon perdit quatre unités de sa flotte de porte-avions et avec eux la plus grande partie de sa force de frappe, pour le maigre gain de deux îles des Aléoutiennes<sup>41</sup>.

Après Midway, les Japonais commirent une autre erreur dans le Sud-Ouest du Pacifique : comme pour la bataille de la Mer de Corail, le plan n'était pas défectueux, mais leur attaque ne fut ni assez rapide, ni lancée avec une force suffisante. Ce n'était pas le meilleur moyen d'atteindre leurs objectifs que d'envoyer aussi peu de troupes à Guadalcanal et en Nouvelle Guinée avec la presque certitude de les voir submergées par les Américains et les Australiens<sup>42</sup>. Effectivement, après six mois de pertes en augmentation régulière, les Japonais furent forcés de se retirer.

---

<sup>39</sup> Ainsi que l'a fort justement souligné Paul Kennedy à cet égard : « Quand il est impératif d'être vainqueur, il vaut mieux se servir d'une masse que d'un marteau ».

<sup>40</sup> Les plans d'un tel repli ont effectivement été réalisés par les Américains à cette époque en cas d'offensive japonaise victorieuse sur Hawaii.

<sup>41</sup> Modelski (G.), Thompson (W. R.), *Seapower in Global Politics 1494-1993*.

<sup>42</sup> A cette remarque, formulée à l'époque lors d'une discussion stratégique, le Général Sugiyama aurait répondu qu'un soldat japonais valait bien 100 Américains et que les qualités principales de ces derniers étaient leur propension à rester en vie - en se rendant - et la vitesse de leur course.



La première phase de la guerre du Pacifique avait vu les Japonais conquérir aisément leurs objectifs initiaux. La deuxième phase les vit faire une tentative pour s'étendre au-delà et finalement subir un arrêt décisif. Il est clair que la troisième phase a vu les Japonais revenir à leur idée d'avant-guerre : tenir solidement dans une zone défensive puissamment fortifiée composée de bases insulaires avec l'espoir de repousser les attaques américaines. En novembre 1942, le Grand Quartier général impérial avait ordonné avec optimisme le maintien et la défense de points stratégiques en Nouvelle Guinée et aux Salomon. Pour permettre d'atteindre ces objectifs, on modifia les structures du commandement régional et on préleva des troupes et des avions sur d'autres îles pour les y diriger... sans grand effet<sup>43</sup>.

Même durant cette phase de la guerre, il existait des indices que la stratégie défensive échouerait. Le Grand Quartier général impérial ou du moins l'Etat-Major général de l'armée avaient les yeux tournés vers le continent asiatique et les renforts envoyés dans le Pacifique Sud et Ouest s'avéraient nettement insuffisants. Tôt en 1943, les forces terrestres et aériennes de Mac Arthur étaient déjà supérieures à celles du Général Imamura et le fossé entre elles se creusait semaine après semaine. La marine recevait également de nombreux renforts, tant dans le Sud-Ouest du Pacifique qu'à Pearl Harbour, où les forces du Centre Pacifique étaient en cours de reconstitution. Les porte-avions japonais et les équipages de l'aviation embarquée ne pouvaient être comparés en nombre et en qualité à leurs homologues américains. Au vu de la montée en puissance des troupes américaines, il était difficile d'imaginer quel choix stratégique ou tactique pourrait arrêter longtemps leur avance.

Ainsi, en 1944, malgré la résistance acharnée des Japonais, Mac Arthur et Halsey purent progresser vers les Salomon, briser la barrière des îles Bismark et avancer le long des côtes de Nouvelle-Guinée. Même avant cela, le Grand Quartier général impérial s'était rendu compte de la tournure que prenaient les événements : en septembre 1943, il avait élaboré une nouvelle stratégie dans laquelle il créait une « zone de défense nationale absolue » s'étendant de la Birmanie à la Malaisie puis à l'Ouest de la Nouvelle Guinée pour atteindre les Carolines et les Marshall et se terminer aux Kouriles. Cette défense inexpugnable devait protéger l'économie de guerre japonaise ainsi que les forces armées pendant la période au cours de laquelle les troupes engagées dans les régions non-vitales se replieraient peu à peu. L'envoi de troupes de Mandchourie et de Chine vers les îles du Pacifique et les Indes orientales devait compléter ce plan<sup>44</sup>.

Cette manœuvre stratégique japonaise était correctement pensée mais les forces mises en œuvre étaient trop faibles. Les Américains possédant la maîtrise de la mer et de l'air pouvaient choisir librement où et quand ils frapperaient et ainsi anéantir à leur gré les garnisons ennemies, exacte réplique de l'action des Japonais au cours

---

<sup>43</sup> Willmott (H. P.), *Victory and Supremacy : From Tulagi to Tarawa. Japanese and Allied Strategies and the Greater East Asia War to November of 1943*, pp. 23-28.

<sup>44</sup> Mais même alors, il restait toujours 15 divisions en Mandchourie et 26 en Chine.

des premiers mois de la guerre<sup>45</sup>. En novembre 1943, les forces américaines du Centre Pacifique s'emparèrent des îles Gilbert et, trois mois plus tard, se saisirent des Marshall et portèrent un coup fatal à la base de Truk dans les Carolines. La « zone de défense nationale absolue » était déjà enfouie et les forces de Mac Arthur s'apprêtaient à agrandir les brèches au Sud<sup>46</sup>.

Depuis sa première défaite aux Salomon, la marine japonaise avait tenté de préserver ses forces en vue d'une contre-attaque décisive pour arrêter la ruée américaine. Une occasion favorable pour une telle victoire se présenta en juin 1944 quand une flotte américaine d'invasion s'approcha des Mariannes, position essentielle dans le système de défense japonais. L'Amiral Toyoda, commandant en chef de la flotte combinée, conçut une stratégie fondamentalement correcte : il essaya de prendre les Américains dans un mouvement de tenaille entre la première flotte mobile et les avions basés sur les îles eux-mêmes ; mais les forces américaines étaient trop puissantes. Dans la bataille des Philippines, la force de Mitscher, composée de quinze porte-avions, détruisit d'abord une des branches de la tenaille en paralysant les bases aériennes insulaires, puis, restant prudemment sur la défensive, mit hors de combat les porte-avions japonais imprudemment avancés et enfin mit en fuite avec de lourdes pertes la première flotte mobile. La défaite des Japonais entraîna la chute des Mariannes et du gouvernement Tojo.

Sans l'appoint suffisant de porte-avions et d'avions, les Japonais ne pouvaient défendre avec succès les Philippines, prochain objectif des Américains ; révisant leur stratégie une nouvelle fois, les chefs de la marine japonaise conçurent un nouveau plan très original : utiliser les porte-avions comme appât et les cuirassés comme force de frappe. Ce plan surprenant réussit pourtant en attirant la flotte d'Halsey - qui couvrait le débarquement du golfe de Leyte - mais les Japonais ne bénéficièrent pas de cette situation avantageuse, le groupe de cuirassés se heurtant par hasard à une force américaine écrasante et l'autre groupe étant victime des indécisions de son chef, payant chèrement son rôle d'appât. L'armée japonaise commit également une erreur en prenant, sur la foi de rapports exagérés concernant les pertes américaines, la décision de résister à Leyte au lieu de concentrer leur défense de l'archipel sur l'île principale de Luzon. Il en résulta la perte de quatre divisions à Leyte et l'affaiblissement de la défense de Luzon. La marine japonaise fut décimée et les troupes des Philippines contraintes de se retrancher dans les montagnes de l'intérieur des terres ; la perte de cet immense archipel interrompit la liaison avec la partie méridionale des possessions du Japon<sup>47</sup>.

Cette perte ne faisait que confirmer un fait qui apparaissait de plus en plus clairement : la défaite menaçait le Japon. Cette éventualité fut très mal reçue par les

---

<sup>45</sup> Hayes (G. P.), *The History of the Joint Chiefs of Staff in World War II. The War Against Japan*, pp. 82-86.

<sup>46</sup> Dunnigan (J. F.), Nofi (A. A.), *Victory at Sea. World War II in the Pacific*, pp. 136-165.

<sup>47</sup> Preston (A.), *Navires et Combats 1939-1945*, pp. 98-107.

chefs de guerre japonais, fortement marqués par le *bushido* : ils se hâtèrent d'ordonner le renforcement des défenses d'Iwo Jima, pensant à juste titre que ce serait le prochain objectif des Américains. La lutte pour la possession de cette île fut des plus meurtrière car les forces japonaises occupaient des positions bien fortifiées qu'elles défendirent pied à pied ; mais comme les Américains possédaient la maîtrise de la mer et de l'air ainsi que des troupes et de l'artillerie en nombre supérieur, l'issue de la lutte ne faisait guère de doute. Au moment de l'attaque américaine sur Okinawa, le Japon, poussé dans ses derniers retranchements, privé de pétrole et surclassé sur tous les plans, mit tous ses espoirs dans les fameux *kamikazes*. Mais bien que ces véritables missiles pilotés aient fait subir de graves dommages aux bâtiments de guerre alliés, leur action était insuffisante pour remettre en question l'offensive. Le Japon qui avait tout parié sur une victoire rapide et une paix négociée avait joué son dernier atout et perdu. Un symbole qui en dit long est le voyage suicidaire du cuirassé Yamato qui ne fut guère qu'un geste gratuit.

Comme nous l'avons évoqué, la décision stratégique la plus désastreuse qu'ont pu prendre les seigneurs de la guerre de Tokyo a été le maintien d'une puissante force en Mandchourie. Pendant la période cruciale qui correspondait aux premières années de la guerre du Pacifique, un dixième de ces 700000 soldats envoyé en Nouvelle Guinée et aux Salomon aurait suffi pour faire pencher la balance en faveur des Japonais. En 1944 et 1945, quand il devint évident que l'assaut des Américains était le principal danger pour le Japon, des troupes de Mandchourie furent envoyées en hâte dans les îles sous mandat, aux Philippines et même au Japon. Mais il était déjà trop tard ; pire, les Russes pouvant dorénavant distraire des troupes de leur front occidental en profitèrent alors pour attaquer et tailler en pièces le reliquat des troupes japonaises en Mandchourie<sup>48</sup>.

Pendant toute cette guerre, la Chine n'a cessé d'être le lieu de l'enlèvement de vastes quantités de troupes japonaises. En 1942 et 1943, de nombreuses offensives intéressant des centaines de milliers de soldats ont permis au Japon d'occuper de grandes étendues du territoire chinois, mais pas la reddition du gouvernement ou de l'armée nationale soutenus par les alliés depuis les Indes. Au cours de l'été 1944, la « Force expéditionnaire de Chine » lança sa plus violente offensive en direction des bases aériennes américaines dans le Sud-Est de la Chine. Dès la fin de l'année, les troupes japonaises réussirent à faire leur jonction avec les forces d'occupation de l'Indochine française et coupèrent ainsi en deux le territoire chinois, en pure perte. Eloignées des centres d'approvisionnement, affrontant une guerre de partisans sur ses arrières et une armée chinoise bien équipée et motivée, ce million d'hommes même ne pouvait suffire à la tâche.

Quant à la stratégie en Birmanie après mai 1942, celle d'une défense statique le long de la frontière indienne, elle était des plus sages puisque le gros des forces aériennes japonaises avait été retiré et que la troupe était trop faible pour envisager une offensive en règle. En 1944 cependant, le Grand Quartier général impérial fut circonvenu par le Lieutenant-Général Mutaguschi : il consentit à ce que la XV<sup>e</sup> armée s'avance vers la plaine d'Imphal. Les motifs de cette opération étaient

---

<sup>48</sup> Wylie (J. C.), *"Reflections on the War in the Pacific"*, pp.82-86.

d'abord de devancer une offensive britannique imminente et d'occuper ensuite une ligne de défense plus sûre afin d'écarter les expéditions des Chindits<sup>49</sup>. Cependant, seuls ceux qui voulaient prendre leurs désirs pour des réalités pouvaient espérer que les trois divisions mal approvisionnées de Mutaguschi seraient en mesure, sans avoir la maîtrise aérienne, d'écraser une force anglo-indienne plus forte et mieux équipée. Les Japonais supportèrent les conséquences de leur imprudence lors des batailles acharnées d'Imphal et de Kohima. Leur emprise sur la Birmanie s'affaiblit en conséquence et Slim profita pleinement de cet avantage en 1945 en jetant ses troupes à travers les lignes ennemies désorganisées, ouvrant ainsi la route de la Chine et délivrant Rangoon.

En outre, le Grand Quartier général impérial commit plusieurs fautes stratégiques par omission dont l'importance fut au moins aussi importante qu'une stratégie opérationnelle fautive. Bien que la guerre ait été totale du côté américain, les chefs japonais sont restés focalisés sur les batailles terrestres et navales, sans prendre en compte les autres aspects de la lutte<sup>50</sup>. Il apparaît qu'ils ont montré peu d'intérêt pour les aspects économiques du conflit une fois leur première offensive passée : ils n'ont pas protégé leurs navires marchands<sup>51</sup>, que ce soit par la technique du convoi, des mesures anti-sous-marines, des portes d'avions d'escorte ou encore des groupes de chasse anti-sous-marins, toutes techniques employées par les alliés en la matière et alors que le Japon était bien plus dépendant de ces transports que tout autre belligérant<sup>52</sup>. Les directives de défense anti-aérienne mirent longtemps à être définies et le remplacement du chasseur Zero ne fut envisagé que tardivement tandis que l'absence d'un bombardier lourd se faisait sentir. Enfin, ils n'ont guère pris conscience du rôle véritable des cuirassés dans un conflit moderne, permettant aux alliés d'exploiter ces lacunes pour renforcer encore le déséquilibre des industries et moyens techniques en présence.

Au cours de l'été 1945, la situation à laquelle faisait face le Japon était purement désespérée : en Asie du Sud-Est, les Britanniques s'étaient emparés de la Birmanie et poursuivaient leur offensive vers Singapour ; les Australiens avaient pris Bornéo et nettoyaient les diverses poches de résistance japonaise. En Chine, la force expéditionnaire japonaise reculait peu à peu en abandonnant le terrain conquis lors de l'offensive de 1944 ; les Soviétiques accentuaient leur concentration de troupes sur la frontière Mandchoue tandis que les Américains avaient reconquis les principales possessions japonaises dans le Pacifique : Okinawa, Iwo Jima, les Philippines, les Mariannes, la Nouvelle-Guinée et les Salomon et, ce faisant, avaient détruit les forces navales japonaises. Les opérations de bombardement stratégique

---

<sup>49</sup> Troupes d'élite britanniques opérant derrière les lignes japonaises.

<sup>50</sup> Evans (D. C.), Peattie (M. R.), *Kaigun. Strategy, Tactics, and Technology in the Imperial Japanese Navy 1887-1941*, pp. 265-282.

<sup>51</sup> Winton (J.), *Convoy. The Defence of the Sea Trade*.

<sup>52</sup> Coutau-Bégarie (H.) (dir.), *L'Évolution de la pensée navale*.

ravageaient le sol japonais, tandis que des opérations de mouillage de mines par sous-marin ou par avion coupaient les approvisionnements dont dépendait la survie du Japon. Les plans et les préparatifs en vue d'une invasion massive par les Alliés de l'archipel japonais étaient déjà en cours, menace que le Japon ne pourrait contrer selon toutes probabilités<sup>53</sup>.

Au vu de ces différents éléments, la seule décision stratégique raisonnable semblait être la capitulation, d'autant que le Japon avait à l'époque, en la personne de l'Amiral Suzuki, Premier Ministre depuis 1945, un partisan de la paix, renforcé par l'adhésion à ses vues des Ministres des Affaires Etrangères et de la Marine et surtout de l'Empereur. La seule inquiétude qui semblait retenir ce parti était l'absence de garanties de la part des Alliés de la continuité de la dynastie impériale.

L'Armée quant à elle s'opposait à ces vues : fidèle à la tradition guerrière japonaise, elle écartait toute idée de capitulation. Comment le Japon, qui n'avait jamais auparavant subi la moindre défaite militaire, pourrait-il capituler alors qu'il disposait encore de millions de soldats et de milliers d'appareils pouvant servir de *kamikazes* ?

Conduits par le Ministre de la Guerre Anami et les Chefs d'Etat-Major, le général Umezumi et l'Amiral Toyoda, les militaires possédaient une stratégie de rechange qui, assez curieusement, reposait sur le principe même d'une invasion alliée : ils espéraient lors de ce débarquement asséner d'énormes pertes aux Alliés, au prix il est vrai d'innombrables morts du côté japonais, de telle sorte que le Japon puisse obtenir la capitulation conditionnelle. Ces chefs de guerre souhaitaient non seulement la continuité de la dynastie impériale, mais aussi le maintien du statut des militaires au Japon et repoussaient l'idée d'une armée d'occupation. Même à ce moment, il s'agissait plus de rêve que de réalité, sinon au niveau des combats, au niveau des résultats diplomatiques.

Même les terribles dommages causés par la bombe d'Hiroshima ne modifièrent en rien ces vues. Les « faucons » du Conseil Suprême de la Guerre, en dépit des souhaits de l'Empereur, refusèrent une entrevue le 8 août. Le lendemain, les Russes déclaraient la guerre au Japon et pénétraient en Mandchourie. Une fois encore, le Ministre de la Guerre et les Chefs d'Etat-Major insistaient sur leur refus d'accepter une armée d'occupation, de désarmer leurs troupes ou de juger les crimes de guerre. Cette décision ne fut en rien modifiée par l'imminence d'une deuxième frappe atomique, ainsi que le démontre leur position lors de la Conférence impériale du soir du 9 août. C'est donc l'Empereur qui, rompant avec la tradition, prit la décision de mettre fin à la guerre, même si le Grand Quartier général impérial fut le siège de graves dissensions jusqu'au 14 août, certains des militaires préférant voir la nation sombrer dans le néant plutôt que d'admettre leur défaite.

Avec le recul actuel, on est en droit de supposer que, quels que fussent l'acharnement et l'habileté avec lesquels les Japonais combattirent, notamment durant les premiers mois du conflit, ils n'auraient pu éviter d'être vaincus par

---

<sup>53</sup> Winton (J.), *Convoy. The Defence of the Sea Trade, 1890-1990*.

l'extraordinaire potentiel industriel américain lors d'un conflit de longue durée. Mais ce résultat fut aussi accéléré, sinon causé, par une longue suite de décisions stratégiques mal pensées et aux conséquences catastrophiques. En premier lieu, l'intérêt exagéré que l'Armée a montré pour le continent asiatique, entraînant la méconnaissance de l'incroyable vulnérabilité de leur flanc exposé à la menace américaine jusqu'au moment où il fut trop tard pour réagir efficacement. Ensuite, la suite de fautes stratégiques commise immédiatement après leur plus glorieuse série de victoires : les échecs de la Mer de Corail, de Midway et de Guadalcanal. Le « sort », qui penchait auparavant résolument pour le Japon, hésita pendant quelques mois que les Américains mirent à profit pour réorganiser leurs troupes et mettre leur puissante économie sur le pied de guerre. Désormais, les Japonais étaient surpassés numériquement tant sur terre que sur les mers et dans les airs. Même les mesures désespérées qu'exigeait la situation japonaise à la fin du conflit furent prises avec retard et trop de parcimonie, tandis que les drames d'Hiroshima et de Nagasaki venaient conclure dans l'horreur un conflit des plus sanglants.

# Bibliographie

## ***Informatique***

Bapat (S.), *Object-Oriented Networks, models for architecture, operations and management*, Prentice-Hall, 1994.

Booch (G.), *Conception orientée objets et applications*, Addison-Wesley, 1992.

Grand (M.), *Java: Language Reference*, O'Reilly, 2<sup>ième</sup> édition, 1997.

McConnell (S.), *Code Complete*, A practical handbook of Software construction, Microsoft Press, 1993.

Perry (W.), *Effective Methods for Software Testing*, John Wiley & Sons, 1995.

Pressman (R. S.), *Software engineering, a practitioner's approche*, third edition, McGraw-Hill., 1992.

Pour une documentation sur Java, le site de sun : <http://java.sun.com/>

Pour des standards, le site du World Wide Web Consortium : <http://www.w3.org/>

Pour des définitions ou acronymes, FOLDOC : <http://wombat.doc.ic.ac.uk/>

## ***Deuxième guerre mondiale***

Bauer (E.), *Histoire Controversée de la Deuxième Guerre mondiale*, Paris, Atlas, 1979, 10 volumes.

Dull (P. S.), *A Battle History of the Imperial Japanese Navy 1941-1945*, Annapolis, Naval Institute Press, 1978.

Dunnigan (J. F.), Nofi (A. A.), *Victory at Sea. World War II in the Pacific*, New York, Morrow, 1995.

Evans (D. C.), Peattie (M. R.), *Kaigun. Strategy, Tactics, and Technology in the Imperial Japanese Navy 1887-1941*, Annapolis, Naval Institute Press, 1997.

Hayes (G. P.), *The History of the Joint Chiefs of Staff in World War II. The War Against Japan*, Annapolis, Naval Institute Press, 1982.

Keegan (J.), *Atlas de la Seconde Guerre mondiale*, Paris, Larousse, 1990.

Masson (P.), *Précis d'histoire de la Seconde Guerre mondiale*, Paris, Tallandier, 1992.

Preston (A.), *Navires et Combats 1939-1945*, Grande-Bretagne, PML editions, 1994.

Purnell (C.) (dir), *La Deuxième Guerre mondiale*, Montréal, édition Franson, 1966

Willmott (H. P.), *Victory and Supremacy : From Tulagi to Tarawa. Japanese and Allied Strategies and the Greater East Asia War to November of 1943*, Annapolis, Naval Institute Press, 1999.

Wylie (J. C.), "Reflections on the War in the Pacific", in *U.S. Naval Institute Proceedings*, avril 1952.

## **Stratégie**

Booth (K.), Herring (E.), *Strategic Studies*, Londres, Mansell, 1994.

Charnay (J.-P.), *Critique de la stratégie*, Paris, L'Herne, 1990.

Charnay (J.-P.), *La Stratégie*, Paris, Presses universitaires de France, Que sais-je ?, 2937, 1995.

Cimbala (S. J.), "Military Strategy", in *The Journal of Slavic Military Studies*, vol. 10, septembre 1997.

Coutau-Bégarie (H.) (dir.), *L'Évolution de la pensée navale*, 7 vols, Paris, FEDN puis ISC-Économica, 1991-1999.

David (C.-P.) (dir), *Les études stratégiques. Approches et concepts*, Paris-Montréal, FEDN-Éditions du Méridien, 1989.

Howard (M.), *The Causes of War*, Londres, Temple Smith, 1983.

Hughes (W. P.), "Naval Tactics and their Influence on Strategy", in *Naval War College Review*, janvier-février 1986.

Luttwak (E. N.), *Le Paradoxe de la stratégie*, Paris, Odile Jacob, 1989.

Luttwak (E. N.), *Strategy and History*, New Brunswick-Oxford, Transaction Books, 1985.

Modelski (G.), Thompson (W. R.), *Seapower in Global Politics 1494-1993*, Londres, Macmillan, 1988.

Phélizon (J.-F.), *L'Action stratégique*, Paris, Économica, Stratèges et stratégie, 1998.

Winton (J.), *Convoy. The Defence of the Sea Trade, 1890-1990*, Londres, Michael Joseph, 1983.